

CSE545

빅데이터처리 및 실습

(Big Data Processing and Practice)

Practice Part 2: Hadoop

담당교수: 전강욱(컴퓨터공학부)

kw.chon@koreatech.ac.kr

개요

- 실습명

- 하둡 설치 및 개발 환경 구축

- 목표

- 하둡을 설치하고, 기본적인 명령어 학습

Java 설치

■ Java (jdk 1.8이상) 설치

- `$ sudo apt-get update`
- `$ sudo apt-get install openjdk-8-jdk`
- `$ java -version`

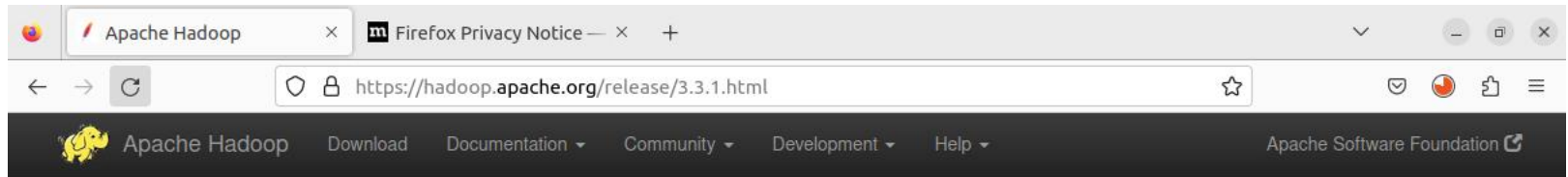
■ JAVA_HOME 설정

- Java 경로 확인
 - `$readlink -f $(which java)`
 - 예: `/usr/lib/jvm/java-8-openjdk-amd64/bin/java`
- Java 경로를 등록
 - `$sudo vim /etc/profile`
 - 아래 내용(예시)을 `/etc/profile` 에 붙여 넣기
 - `Export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64`
 - `Export PATH=$PATH:$JAVA_HOME/bin`
 - `$source /etc/profile`

Hadoop 설치

■ Hadoop 다운로드

□ <https://Hadoop.apache.org/release/3.3.1.html>



Release 3.3.1 available

This is the first stable release of Apache Hadoop 3.3.x line. It contains 697 bug fixes, improvements and enhancements since 3.3.0.

Users are encouraged to read the [overview of major changes](#) since 3.3.0. For details of 697 bug fixes, improvements, and other enhancements since the previous 3.3.0 release, please check [release notes](#) and [changelog](#) detail the changes since 3.3.0.

2021 Jun 15

[Download tar.gz](#)

[\(checksum signature\)](#)

[Download aarch64 tar.gz](#)

[\(checksum signature\)](#)

[Download src](#)

[\(checksum signature\)](#)

[Documentation](#)

Apache Hadoop, Hadoop, Apache, the Apache feather logo, and the Apache Hadoop project logo are either registered trademarks or trademarks of the Apache Software Foundation in the United States and other countries

Copyright © 2006-2023 The Apache Software Foundation

[Privacy policy](#)



THE **APACHE**® SOFTWARE FOUNDATION
<http://www.apache.org/>

Hadoop 설치 (계속)

- 하둡 설치 디렉토리는 **"/home/Hadoop/Install"**로 가정
- 1. 압축 풀기
 - `$ tar -xvfz /home/Hadoop/Install/Hadoop-3.3.1.tar.gz`
- 2. Hadoop 설정 디렉토리로 이동
 - `$ cd /home/Hadoop/Install/Hadoop-3.3.1/etc/Hadoop`
- 3. Hadoop 설정 파일(hadoop-env.sh) 수정 (예제)
 - `$ export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/bin/java`
 - `$ export HADOOP_HOME= /home/Hadoop/Install/Hadoop-3.3.1`
 - `$ export HADOOP_CONF= /home/Hadoop/Install/Hadoop-3.3.1/etc/hadoop`

Hadoop 설치 (계속)

■ 4. 설정 파일 수정 (core-site.xml)

```
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://localhost:9000</value>
  </property>
</configuration>
```

■ 5. 설정 파일 수정 (hdfs-site.xml)

```
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>
</configuration>
```

Hadoop 설치 (계속)

■ 6. 설정파일 수정 (mapred-site.xml)

```
<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
  <property>
    <name>mapreduce.application.classpath</name>
    <value>$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/*:$
HADOOP_MAPRED_HOME/share/hadoop/mapreduce/lib/*
    </value>
  </property>
</configuration>
```

Hadoop 설치 (계속)

■ 7. 설정파일 수정 (yarn-site.xml)

```
<configuration>
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
  <property>
    <name>yarn.nodemanager.env-whitelist</name>
    <value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP
P_HDFS_HOME,
HADOOP_CONF_DIR,CLASSPATH_PREPEND_DISTCAC
HE,HADOOP_YARN_HOME,HADOOP_MAPRED_HOME
    </value>
  </property>
</configuration>
```


Hadoop 설치 (계속)

■ 8. SSH 설정

- ❑ `$ sudo apt-get install openssh-server`
- ❑ `$ ssh-keygen -t rsa -P "" -f ~/.ssh/id_rsa`
- ❑ `$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys`
- ❑ `$ chmod 0600 ~/.ssh/authorized_keys`
- ❑ `$ ssh localhost`
 - 접속 확인

Hadoop 설치 (계속)

- 9. NameNode (HDFS 메타 데이터 관리) 포맷
 - `$ hdfs namenode -format`
 - NameNode를 포맷하면 `dfs.namenode.name.dir` 경로의 `fsimage`와 `edits` 파일을 초기화 시킴
 - 설치 후 최초 1회만 수행 함

Hadoop 설치 (계속)

■ 10. HDFS 데몬 실행

❑ \$ sbin/start-dfs.sh

```
hadoop@hadoop-VirtualBox:~/Install/hadoop-3.3.1$ ./sbin/start-dfs.sh
```

```
Starting namenodes on [localhost]
```

```
Starting datanodes
```

```
Starting secondary namenodes [hadoop-VirtualBox]
```

```
hadoop-VirtualBox: Warning: Permanently added 'hadoop-virtualbox' (ED25519) to the list of known hosts.
```

```
hadoop@hadoop-VirtualBox:~/Install/hadoop-3.3.1$ jps
```

```
29697 SecondaryNameNode
```

```
29382 NameNode
```

```
29803 Jps
```

```
29500 DataNode
```

HDFS 데몬 실행

jps 명령어를 통해서 Java Process 확인

Overview 'localhost:9000' (✓active)

Started:	Sat Aug 26 19:29:45 +0900 2023
Version:	3.3.1, ra3b9c37a397ad4188041dd80621bdeefc46885f2
Compiled:	Tue Jun 15 14:13:00 +0900 2021 by ubuntu from (HEAD detached at release-3.3.1-RC3)
Cluster ID:	CID-b8f96995-8903-4c81-b39a-1539dff80aa3
Block Pool ID:	BP-1855325328-127.0.1.1-1693045718158

Summary

Security is off.
Safemode is off.
1 files and directories, 0 blocks (0 replicated blocks, 0 erasure coded block groups) = 1 total filesystem object(s).
Heap Memory used 41.03 MB of 61.54 MB Heap Memory. Max Heap Memory is 945.44 MB.

NameNode의 Web Interface 가 접속이 잘되면 성공
- <http://localhost:9870/>

Hadoop 설치 (계속)

■ 11. YARN 데몬 실행

❑ \$ sbin/start-yarn.sh

```
hadoop@hadoop-VirtualBox:~/Install/hadoop-3.3.1$ sbin/start-yarn.sh
```

```
Starting resourcemanager
```

```
Starting nodemanagers
```

```
hadoop@hadoop-VirtualBox:~/Install/hadoop-3.3.1$ jps
```

```
30369 Jps
```

```
29697 SecondaryNameNode
```

```
29382 NameNode
```

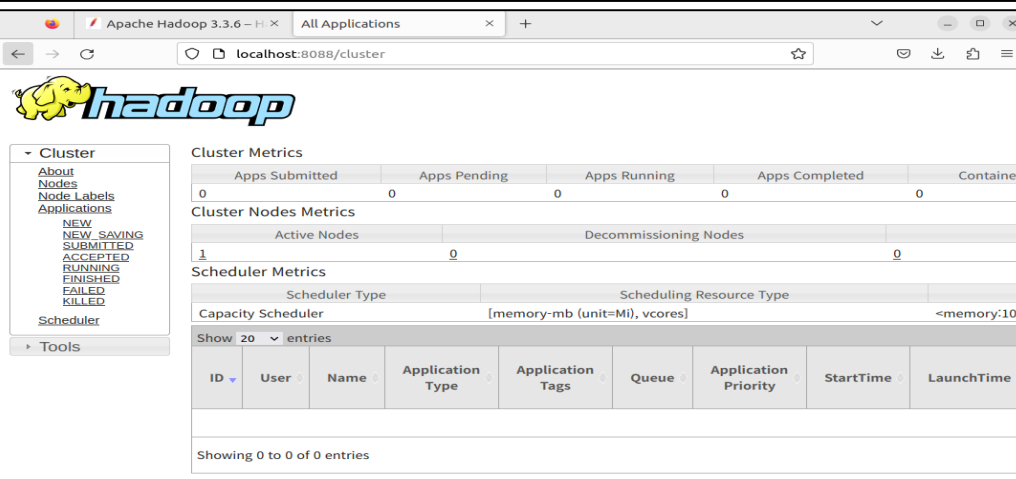
```
29500 DataNode
```

```
29933 ResourceManager
```

```
30046 NodeManager
```

YARN 데몬 실행

jps 명령어를 통해서 Java Process 확인



The screenshot shows the Hadoop Resource Manager Web Interface. The left sidebar contains a navigation menu with options like Cluster, About, Nodes, Node Labels, Applications, NEW, NEW SAVING, SUBMITTED, ACCEPTED, RUNNING, FINISHED, FAILED, KILLED, Scheduler, and Tools. The main content area displays Cluster Metrics, Cluster Nodes Metrics, and Scheduler Metrics. Below these, there is a table showing application entries with columns for ID, User, Name, Application Type, Application Tags, Queue, Application Priority, StartTime, and LaunchTime. The table currently shows 0 entries.

Resource Manager의 Web Interface 가 접속이 잘되면 성공
- <http://localhost:8088>

Hadoop 설치 (계속)

- 하둡 설치 디렉토리는 **"/home/Hadoop/Install"**로 가정
- 12. 예제 프로그램 (word count) 실행
 - **\$ hdfs dfs -mkdir -p /user/hadoop/test**
 - HDFS에 경로 (/user/hadoop/test) 생성
 - **\$ hdfs dfs -put /home/Hadoop/Install/Hadoop-3.3.1/etc/Hadoop/core-site.xml /user/Hadoop/test**
 - Local에 있는 core-site.xml파일을 HDFS에 적재
 - **\$ yarn jar /home/Hadoop/Install/Hadoop-3.3.1/share/hadoop/mapreduce/Hadoop-mapreduce-examples-3.3.1.jar wordcount test output**
 - 테스트용으로 이미 구현된 wordcount예제를 수행

제출물

- Hadoop 설치를 위한 단계별로 캡처화면 생성하여 제출
 - 본인이 한 것임을 증명할 수 있도록 캡처화면 (파일 이름 등)에 본인 학번이 포함

개발환경 구축

담당교수: 전강욱(컴퓨터공학부)

kw.chon@koreatech.ac.kr

Eclipse & Maven 설치

■ Eclipse

- 이클립스 다운로드 페이지: <http://eclipse.org/downloads>
 - 최신 버전 설치
- `$ tar xvfz ./eclipse-inst-jre-linux64.tar.gz`
- `$ cd eclipse-installer`
- `$./eclipse-inst`
- 이후 eclipse for Java Developer 설치

■ Maven

- `$ sudo apt update`
- `$ sudo apt install maven`
- `$ mvn -version`

개발환경 구축

- Eclipse 실행 후 Maven 프로젝트 생성
 - File → New → Maven Project
- pom.xml 수정 (MapReduce Dependency 추가)

```
<dependencies>
  <dependency>
    <groupId>org.apache.hadoop</groupId>
    <artifactId>hadoop-mapreduce-client-core</artifactId>
    <version>3.0.0</version>
  </dependency>
  <dependency>
    <groupId>org.apache.hadoop</groupId>
    <artifactId>hadoop-common</artifactId>
    <version>3.0.0</version>
  </dependency>
  <dependency>
    <groupId>org.apache.hadoop</groupId>
    <artifactId>hadoop-mapreduce-client-jobclient</artifactId>
    <version>3.0.0</version>
  </dependency>
</dependencies>
```

출처:
<https://jangunmp1.github.io/tips/2019/04/17/hadoop.html>

개발환경 구축 (계속)

■ MapReduce 코드 작성 (Package 목록)

```
import java.io.IOException;
import java.util.StringTokenizer;
import org.apache.hadoop.conf.Configured;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;
import org.apache.hadoop.util.Tool;
import org.apache.hadoop.util.ToolRunner;
```

개발환경 구축 (계속)

■ MapReduce 코드 작성 (Main 함수 및 Map 함수)

```
public class WordCount extends Configured implements Tool{
    public static void main(String[] args) throws Exception {
        ToolRunner.run(new WordCount(), args);
    }
    public static class WCMapper extends Mapper<Object, Text, Text, IntWritable>{
        Text word = new Text();
        IntWritable one = new IntWritable(1);
        @Override protected void map(Object key, Text value,
            Mapper<Object, Text, Text, IntWritable>.Context context
            throws IOException, InterruptedException
        {
            StringTokenizer st = new StringTokenizer(value.toString());
            while(st.hasMoreTokens()) {
                word.set(st.nextToken());
                context.write(word, one);
            }
        }
    }
}
```

개발환경 구축 (계속)

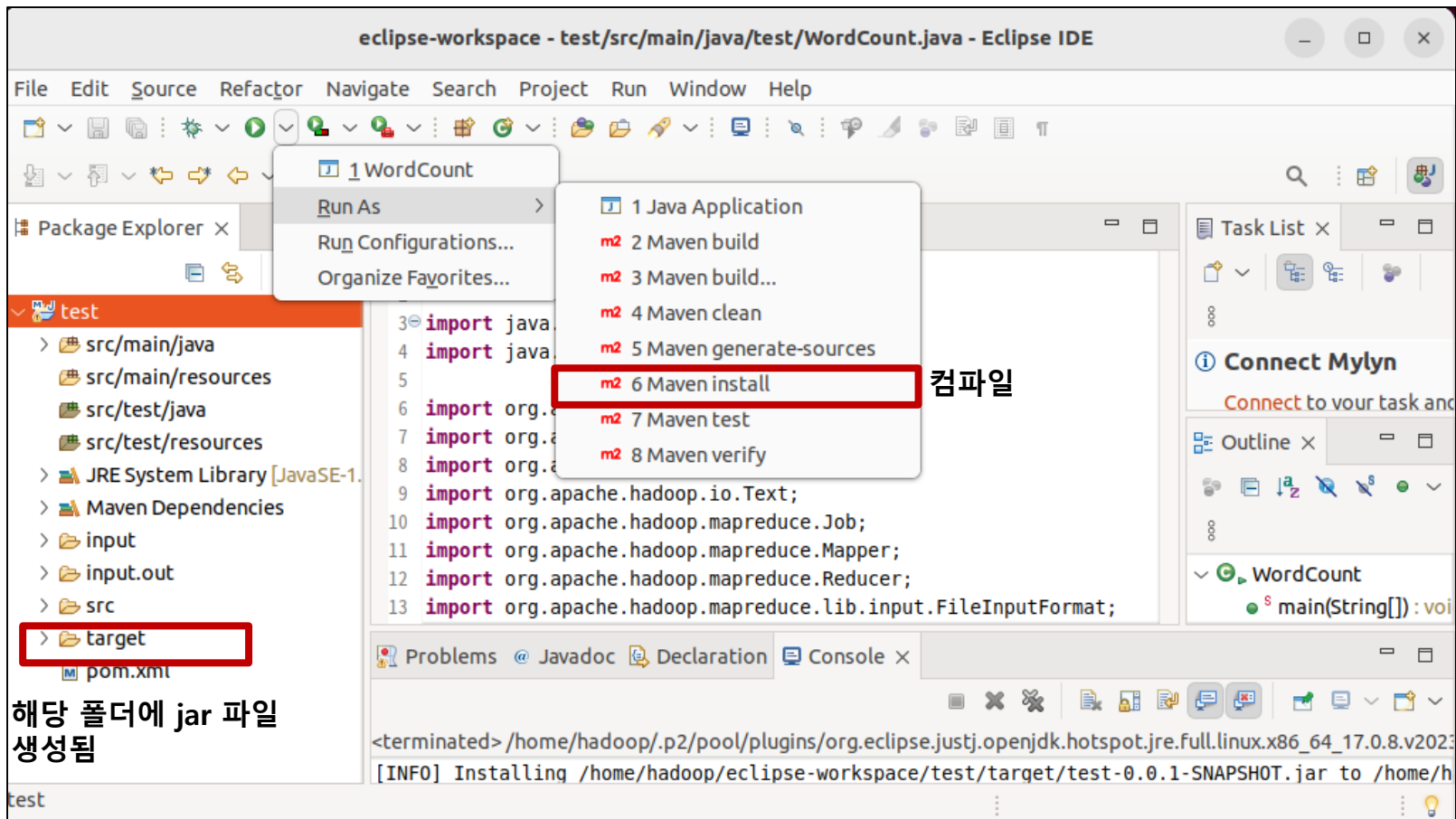
■ MapReduce 코드 작성 (Reduce 함수)

```
public static class WCReducer extends
    Reducer<Text, IntWritable, Text, IntWritable>
{
    IntWritable oval = new IntWritable();
    @Override protected void reduce(Text key, Iterable<IntWritable>
                                     values,
    Reducer<Text, IntWritable, Text, IntWritable>.Context context)
        throws IOException, InterruptedException
    {
        int sum = 0;
        for(IntWritable value : values)
        {
            sum += value.get();
        } oval.set(sum);
        context.write(key, oval);
    }
}
```

개발환경 구축 (계속)

■ Maven 컴파일 (jar파일 생성)

- Eclipse에서 [Run As] → [Maven Install]



해당 폴더에 jar 파일
생성됨

개발환경 구축 (계속)

■ jar 파일 실행 명령어

- `yarn jar ./test-0.0.1-SNAPSHOT.jar test.WordCount /hadoop/test`

Jar File

Main Class

Input Path on HDFS

```
hadoop@hadoop-VirtualBox:~/eclipse-workspace/test/target$ yarn jar ./test-0.0.1-SNAPSHOT.jar test.WordCount /hadoop/test/
2023-08-26 23:56:37,763 INFO client.DefaultNoHARMAFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2023-08-26 23:56:38,445 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/hadoop/.staging/job_1693045913947_0002
2023-08-26 23:56:38,747 INFO input.FileInputFormat: Total input files to process : 1
2023-08-26 23:56:38,831 INFO mapreduce.JobSubmitter: number of splits:1
2023-08-26 23:56:39,068 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1693045913947_0002
2023-08-26 23:56:39,068 INFO mapreduce.JobSubmitter: Executing with tokens: []
2023-08-26 23:56:39,276 INFO conf.Configuration: resource-types.xml not found
2023-08-26 23:56:39,281 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2023-08-26 23:56:40,084 INFO impl.YarnClientImpl: Submitted application application_1693045913947_0002
2023-08-26 23:56:40,160 INFO mapreduce.Job: The url to track the job: http://hadoop-VirtualBox:8088/proxy/application_1693045913947_0002/
2023-08-26 23:56:40,161 INFO mapreduce.Job: Running job: job_1693045913947_0002
2023-08-26 23:56:48,381 INFO mapreduce.Job: Job job_1693045913947_0002 running in uber mode : false
2023-08-26 23:56:48,382 INFO mapreduce.Job: map 0% reduce 0%
2023-08-26 23:56:53,441 INFO mapreduce.Job: map 100% reduce 0%
2023-08-26 23:56:58,481 INFO mapreduce.Job: map 100% reduce 100%
2023-08-26 23:56:58,485 INFO mapreduce.Job: Job job_1693045913947_0002 completed successfully
2023-08-26 23:56:58,603 INFO mapreduce.Job: Counters: 54
File System Counters
FILE: Number of bytes read=5450
```

제출물

- 단계별로 캡처화면 생성하여 제출
 - 본인이 한 것임을 증명할 수 있도록 캡처화면 (파일 이름 등)에 본인 학번이 포함

HDFS 실습

담당교수: 전강욱(컴퓨터공학부)

kw.chon@koreatech.ac.kr

HDFS 명령

- **hdfs dfs *-command -option args***

- e.g., hdfs dfs -mkdir -p /user/hadoop

- **command**

- ls [-d][-h][-R] : 파일 또는 디렉토리 목록
- du [-s][-h] : 파일 용량 확인
- cat, text : 파일 내용 보기
- mkdir [-p] : 디렉토리 생성
- put, get : 파일 복사 (Local <-> HDFS)
- getmerge [-nl] : 병합해서 로컬에 저장(nl은 각 파일 끝에 개행 문자 포함)
- cp, mv : 파일 복사, 이동(HDFS <-> HDFS)
- rm [-R][-skipTrash] : 파일 삭제, 디렉토리 삭제, 완전 삭제
- count [-q] : 카운트 값 조회
- tail : 파일의 마지막 내용 확인
- chmod, chown, chgrp : 권한, 소유주, 그룹 변경

HDFS 명령 (계속)

- **hdfs dfs *-command -option args***
 - e.g., `hdfs dfs -mkdir -p /user/hadoop`
- **command (계속)**
 - `touchz` : 0바이트 파일 생성
 - `stat [-R] <format>` : 통계 정보 조회
 - 포맷 : %b(바이트수) %F(파일인지디렉토리인지) %u(소유주) %g(그룹) %n(이름) %o(블록 크기) %r(복제수) %y(날짜 및 시간) %Y(유닉스타임스탬프)
 - `setrep` : 복제 수 변경
 - `expunge` : 휴지통 비우기
 - `test -[edz]`: 파일 형식 확인(empty, zero, dir)

실습

■ 데이터 준비

- ❑ `$ wget http://stat-computing.org/dataexpo/2009/2007.csv.bz2`
- ❑ `$ wget http://stat-computing.org/dataexpo/2009/2008.csv.bz2`
- ❑ `$ bunzip2 2007.csv.bz2`
- ❑ `$ bunzip2 2008.csv.bz2`

실습 (계속)

- 사용자의 홈 디렉토리를 생성
- 사용자 홈 디렉토리에 airline 디렉토리를 생성
- airline 디렉토리에 2008.csv 파일을 업로드
- airline 디렉토리에 2007.csv 파일을 업로드
- 로컬의 2008.csv 파일을 삭제
- HDFS의 2008.csv 파일을 로컬에 저장
- airline 디렉토리를 삭제
- 루트에 airline 디렉토리를 생성
- /airline 디렉토리에 2008.csv 파일을 업로드
- 2008.csv 파일의 처음 5라인을 출력
- 2008.csv 파일의 마지막 1KB를 출력
- 2008.csv 파일의 통계 정보를 조회
- 2008.csv 파일의 복제 데이터 개수를 변경
- 2008.csv 파일의 복제 수를 확인
- 2008.csv 파일의 복제 수를 1로 변경

제출물

- 단계별로 캡처화면 생성하여 제출
 - 본인이 한 것임을 증명할 수 있도록 캡처화면 (파일 이름 등)에 본인 학번이 포함

실습 데이터 준비: TPC-H & Frequent Itemset Mining & K-Means Clustering

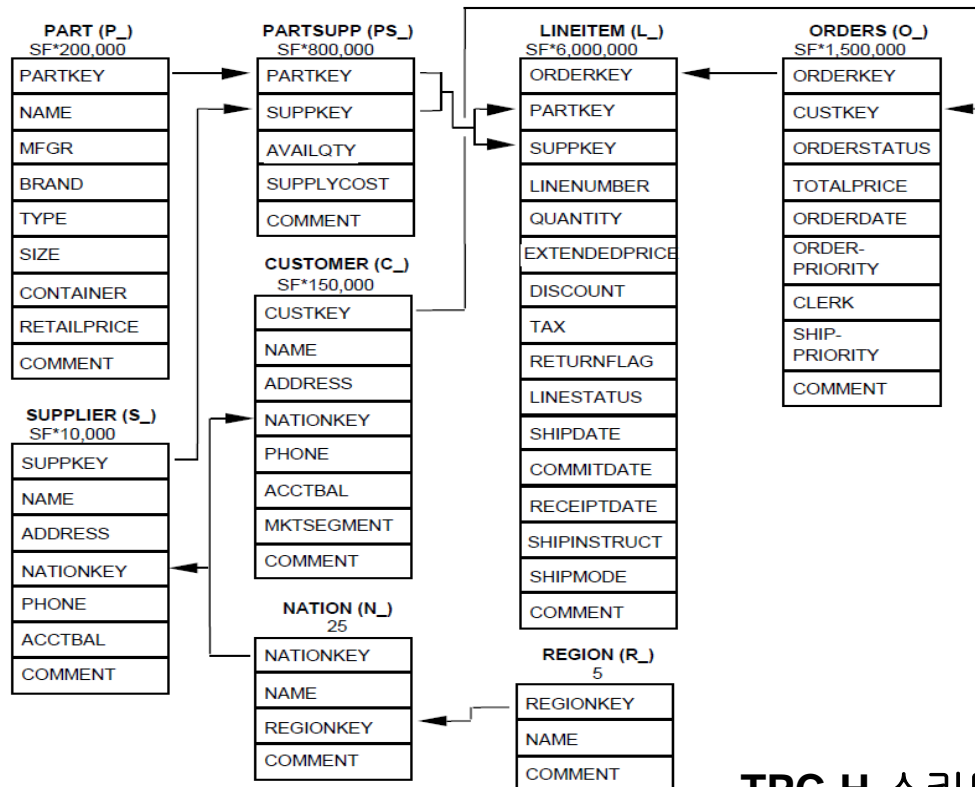
담당교수: 전강욱(컴퓨터공학부)

kw.chon@koreatech.ac.kr

TPC-H 벤치마크

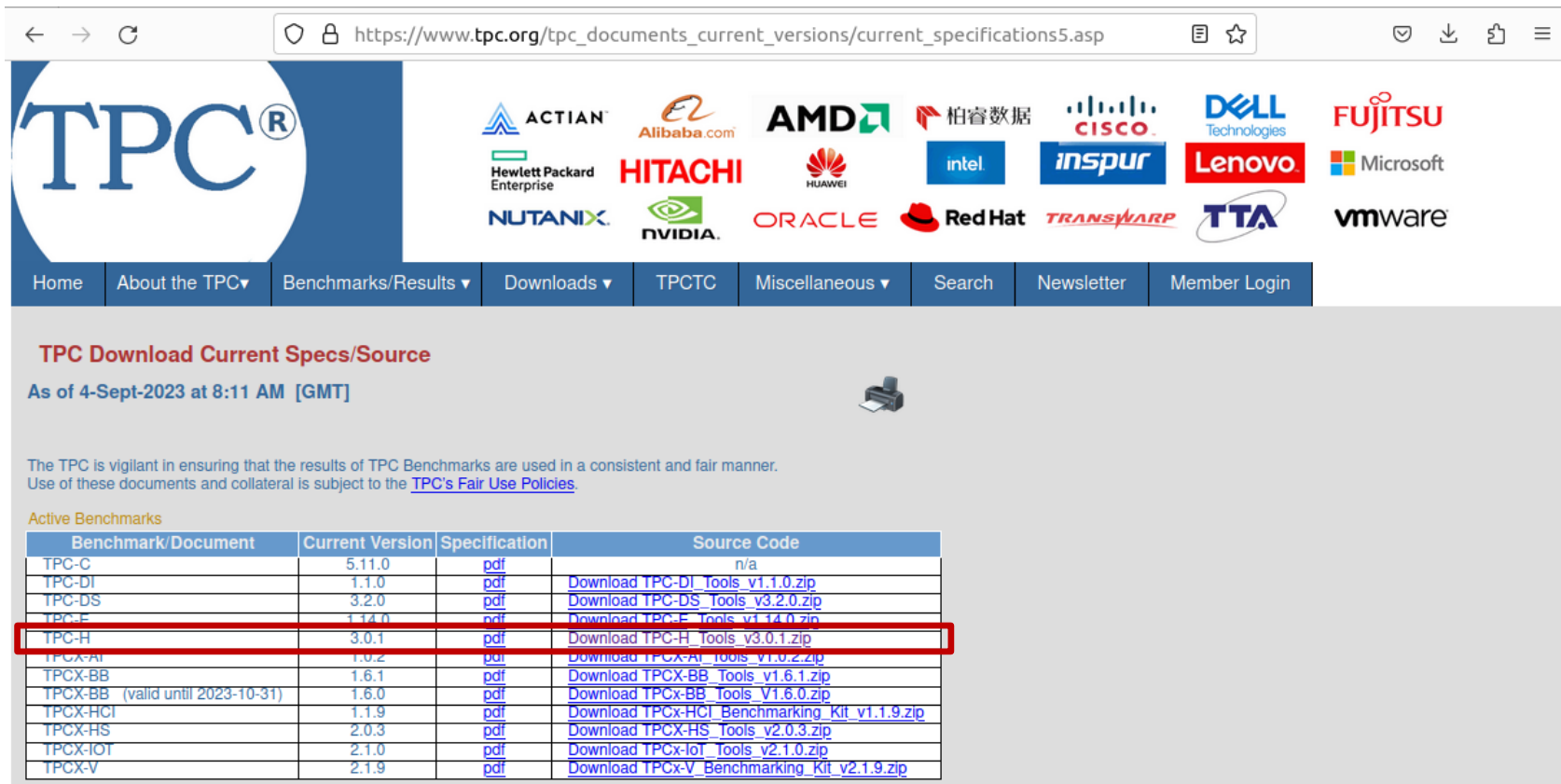
■ DBMS의 성능 측정 시 사용되는 벤치마크

- 의사 결정 용도의 시스템에 대한 성능측정
 - Business를 위한 Adhoc Query (특별한 목적을 위해서) 위주로 구성
- 8개의 테이블로 이루어진 데이터베이스에 대해 22개의 사전 정의된 Query를 제공



TPC-H 다운로드 & 데이터 생성

- TPC-H 웹사이트 접속
 - <http://www.tpc.org>
- 다운로드 페이지로 이동 후 TPC-H 클릭 (아래 그림 확인)
 - Downloads → Downloads programs and Specifications



The screenshot shows the TPC website's 'Downloads' page. The page header includes the TPC logo and a navigation bar with links like Home, About the TPC, Benchmarks/Results, Downloads, TPCTC, Miscellaneous, Search, Newsletter, and Member Login. The main content area is titled 'TPC Download Current Specs/Source' and shows a table of active benchmarks. The 'TPC-H' row is highlighted with a red box.

Benchmark/Document	Current Version	Specification	Source Code
TPC-C	5.11.0	pdf	n/a
TPC-DI	1.1.0	pdf	Download TPC-DI_Tools_v1.1.0.zip
TPC-DS	3.2.0	pdf	Download TPC-DS_Tools_v3.2.0.zip
TPC-E	1.14.0	pdf	Download TPC-E_Tools_v1.14.0.zip
TPC-H	3.0.1	pdf	Download TPC-H_Tools_v3.0.1.zip
TPC-HI	1.0.2	pdf	Download TPC-HI_Tools_v1.0.2.zip
TPCX-BB	1.6.1	pdf	Download TPCX-BB_Tools_v1.6.1.zip
TPCX-BB (valid until 2023-10-31)	1.6.0	pdf	Download TPCX-BB_Tools_V1.6.0.zip
TPCX-HCI	1.1.9	pdf	Download TPCx-HCI_Benchmarking_Kit_v1.1.9.zip
TPCX-HS	2.0.3	pdf	Download TPCX-HS_Tools_v2.0.3.zip
TPCX-IOT	2.1.0	pdf	Download TPCx-IOT_Tools_v2.1.0.zip
TPCX-V	2.1.9	pdf	Download TPCx-V_Benchmarking_Kit_v2.1.9.zip

TPC-H 다운로드 & 데이터 생성(계속)

- 본인 정보입력 및 라이선스 동의 후 다운로드 클릭
- 이후 입력한 Email 주소로 다운로드 링크 수령
(다음 페이지 참조)

TPC-H Tools Download

The TPC Tools are available free of charge, however all users must agree to the licensing terms and register prior to use.
Please download and read the TPC-Tools License Agreement prior to registering for the download.

Ubuntu Software

* First Name
* Last Name
* Company / Affiliation
* Occupation
* Country
* Email
* Terms

(* Required)

Note 1: You will receive an E-mail at the address that you entered above with a link to the files to download
The TPC will not share your E-mail with anybody. - (see TPC's [Privacy Policy](#))
Submitting an invalid E-mail address will result in not being able to download the software.

로봇이 아닙니다.

reCAPTCHA
개인정보 보호 - 약관

☒ I have read and agree to the [TPC End User License Agreement](#) - (.txt file).

Download Cancel

TPC-H 다운로드 & 데이터생성 (계속)

■ Email 내 링크 클릭 후, TPC-H_Tools_v3.01.zip 파일 다운로드

The image shows a two-part process for downloading TPC-H tools. The top part is an email from 'Info@tpc.org' to 'chon0705@gmail.com' with the subject 'TPC-Tools (TPC-H) Download Confirmation'. The email body contains a thank you message and a link to download the software. The link is highlighted with a red box: https://tpc.org/tpc_documents_current_versions/download_programs/tools-download5.asp?email=chon0705@gmail.com&bm_type=TPC-H&bm_version=3.0.1&download_key=5E82BC0A%2DD479%2D4ED8%2D90C8%2D5C4982047541. The bottom part is a screenshot of the web page reached by clicking the link. The page title is 'TPC-H Tools Download' and it says 'Thank you for registering to download the TPC tools software package.' A link to 'TPC-H_Tools_v3.0.1.zip (Tools)' is highlighted with a red box. Below this, there are instructions: 'Please note that some browsers block the automatic download option (e.g.: 'MS Edge'). In that case cut and paste the link from the E-mail that you have received into a different browser (e.g.: 'Google Chrome' or 'Firefox')'. It also mentions that the download might take 30 minutes or more depending on the network connection and file size. Finally, it states that the file can only be downloaded once and provides a link to register again if the download fails.

TPC-Tools (TPC-H) Download Confirmation ▶ 받은편지함 x

Info@tpc.org 도메인: email-od.com 나에게
chon0705@gmail.com

오후 4:31 (42분 전) ☆ ↶ ⋮

Thank you for signing up to download the TPC-H Tools.

Please select the link below or copy and paste it into your web browser to download the software:

https://tpc.org/tpc_documents_current_versions/download_programs/tools-download5.asp?email=chon0705@gmail.com&bm_type=TPC-H&bm_version=3.0.1&download_key=5E82BC0A%2DD479%2D4ED8%2D90C8%2D5C4982047541

Note: A new (temporary) file is being created right now for you. Depending on the size of the file(s) this might take up to 2 minutes.

The temporary file will be available for download for about 3 hours and will be deleted then.

This link will be valid for about three hours for a single download. After that, you will have to register for a new download again.

TPC-H Tools Download
Thank you for registering to download the TPC tools software package.

• [TPC-H_Tools_v3.0.1.zip \(Tools\)](#)

Please note that some browsers block the automatic download option (e.g.: 'MS Edge'). In that case cut and paste the link from the E-mail that you have received into a different browser (e.g.: 'Google Chrome' or 'Firefox').

Depending on your network connection and the size of the file to be downloaded, it might take 30 minutes or even more for the download to finish (TPCx-V - 1.8 GB) - please be patient. Most of the downloads will finish within a few seconds.

The file can only be downloaded once. If you don't see a file to download on this screen, please register again [here](#).

If you don't see a link to download the tools you have requested, please click [here](#).

TPC-H 다운로드 & 데이터 생성 (계속)

■ make 설치

- `$ sudo apt install make`

■ TPC-H 파일 압축 해제 및 파일 생성

- `$ unzip TPC-H-Tool.zip`
 - 다운로드 받은 파일을 압축해제 하는 것이며, 압축파일 명은 다를 수 있음
- `$ cd 'TPC-H V3.0.1'`
- `$ cd dbgen`
- `$ cp makefile.suite Makefile`
- `$ vi Makefile` // Makefile 내 아래 내용 변경
 - `DATABASE = SQLSERVER`
 - `MACHINE = LINUX`
 - `WORKLOAD = TPCH`
 - `CC = gcc`
- `$ make dbgen`
- `$ time ./dbgen`

FIMI

■ FIMI Repository

- Frequent Itemset Mining 알고리즘의 성능 평가를 위하여 많이 사용되는 데이터 및 알고리즘들을 포함
 - 데이터는 트랜잭션 형태의 데이터(데이터 형식은 뒷장 참조)
 - 알고리즘들은 Single PC & Single Thread 기반 C++ 프로그램
- <http://fimi.uantwerpen.be/data/>

데이터 형식

■ 입력 데이터

- 각 트랜잭션이 1개 Line이며, 아이템들이 공백으로 분리되어 있음
 - 아이템들은 Non-Negative 정수
- 예제) [EOF]는 파일의 끝을 나타냄
 - 0 1 2
 - 1
 - 2 3 4

■ 출력 데이터(Frequent Itemset Mining의 결과)

- 여러 개의 Line들로 구성되어 있고, 각 라인은 1개의 frequent itemset과 support(0과 1사이)로 구성
- 예제) 1 2 4는 frequent itemset이며, 전체 데이터베이스에서 50% 나타났다는 것을 가리킴
 - 1 2 4 (0.5)

데이터 다운로드

- 아래 명령어를 통해서 데이터 다운로드 가능하며, 실습 환경을 고려하였을 때 큰 규모 데이터는 수행 불가능하여, 작은 데이터로 실습 예정임
 - \$ wget
<http://fimi.uantwerpen.be/data/T10I4D100K.dat>
 - \$ wget
<http://fimi.uantwerpen.be/data/T40I4D100K.dat>
- FIMI Repository에서 다른 실제 데이터도 구해볼 수 있음

K-Means Clustering

- $\langle x, y \rangle$ 의 Set을 생성
 - 10,000 포인트
 - x, y 는 double형 실수

MapReduce 알고리즘: Reduce-Side Join

담당교수: 전강욱(컴퓨터공학부)

kw.chon@koreatech.ac.kr

조인(Join) 연산

- 조인 연산은 2개 또는 그 이상의 테이블 대상으로 질의할 때 사용됨
 - 테이블들의 특정 컬럼의 관계에 기반하여 연산을 수행

```
SELECT column_name(s)
FROM table_name1, table_name2
ON table_name1.column_name = table_name2.column_name;
```

Table : Grade

Id	Grade
1	A
2	B
3	A

Table : Student

Id	Name
1	John
2	Make
3	Deny

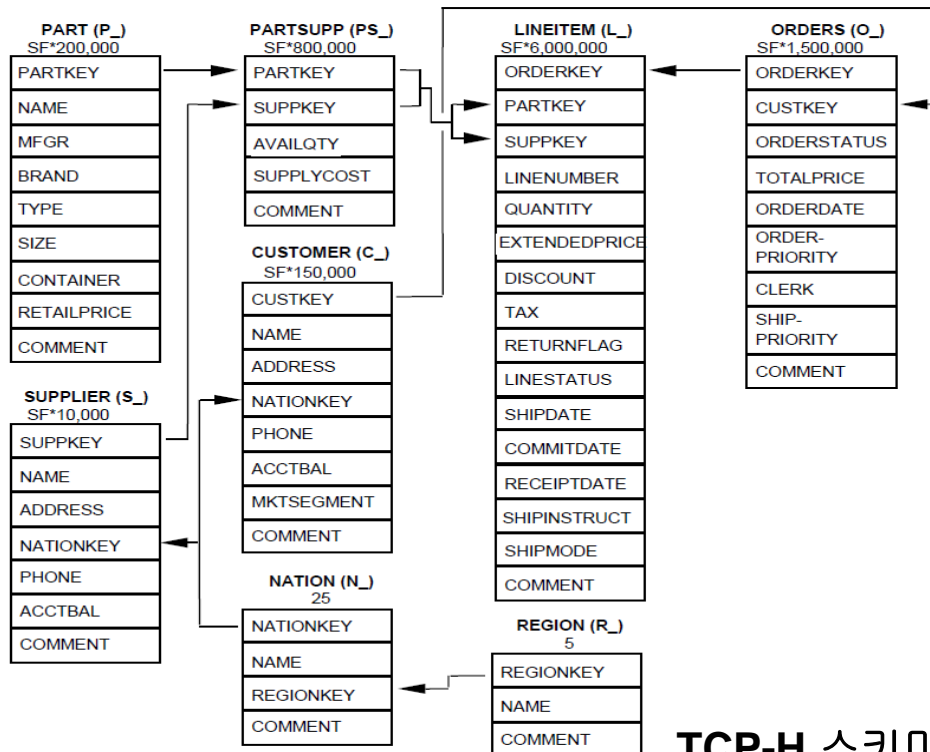
SELECT *
FROM Student, Grade
ON Student.id = Grade.id;



Id	Name	Grade
1	John	A
2	Make	B
3	Deny	A

데이터베이스 응용: TCP-H

- DBMS의 성능을 측정하기 위한 용도의 의사 결정 용(Decision Support) 벤치마크
 - 대규모 데이터 제공 (Scaling 가능)
 - 복잡한 Query 포함 (추후 예제 확인)



TCP-H 스키마.

데이터베이스 기본 연산 리뷰

■ SELECT 문: 테이블에서 데이터를 추출

□ e.g., `SELECT * FROM table_name;`

- * 은 테이블의 전체 컬럼 추출한다는 의미

Table : Grade

Id	Name	Grade
1	John	A
2	Make	B
3	Deny	A
4	Jain	C

`SELECT name FROM Grade;`



Name
John
Make
Deny
Jain

데이터베이스 기본 연산 리뷰(계속)

- **SELECT JOIN문**: 여러 테이블에서 데이터를 추출하기 위한 방법
 - 테이블들 내 컬럼들 간의 특정 관계에 기반하여 데이터를 추출
 - e.g., **SELECT** *col_name(s)* **FROM** *tbl_name1, tbl_name2* **ON** *tbl_name1.col_name = tbl_name2.col_name;*

Table : Grade

Id	Grade
1	A
2	B
3	A

*SELECT * FROM Student, Grade
on Student.id = Grade.id;*



Table : Student

Id	Name
1	John
2	Make
3	Deny

Id	Name	Grade
1	John	A
2	Make	B
3	Deny	A

Q20 in TCP-H

- 잠재적인 부품 프로모션(Potential Part Promotion Query) 질의
- 질의
 - The potential part promotion query identifies suppliers who have an excess of a given part available
 - An excess is defined to be more than 50% of the parts like the given part that the supplier shipped in a given year for a given nation
 - Only parts whose names share a certain naming convention are considered

Q20 in TCP-H (계속)

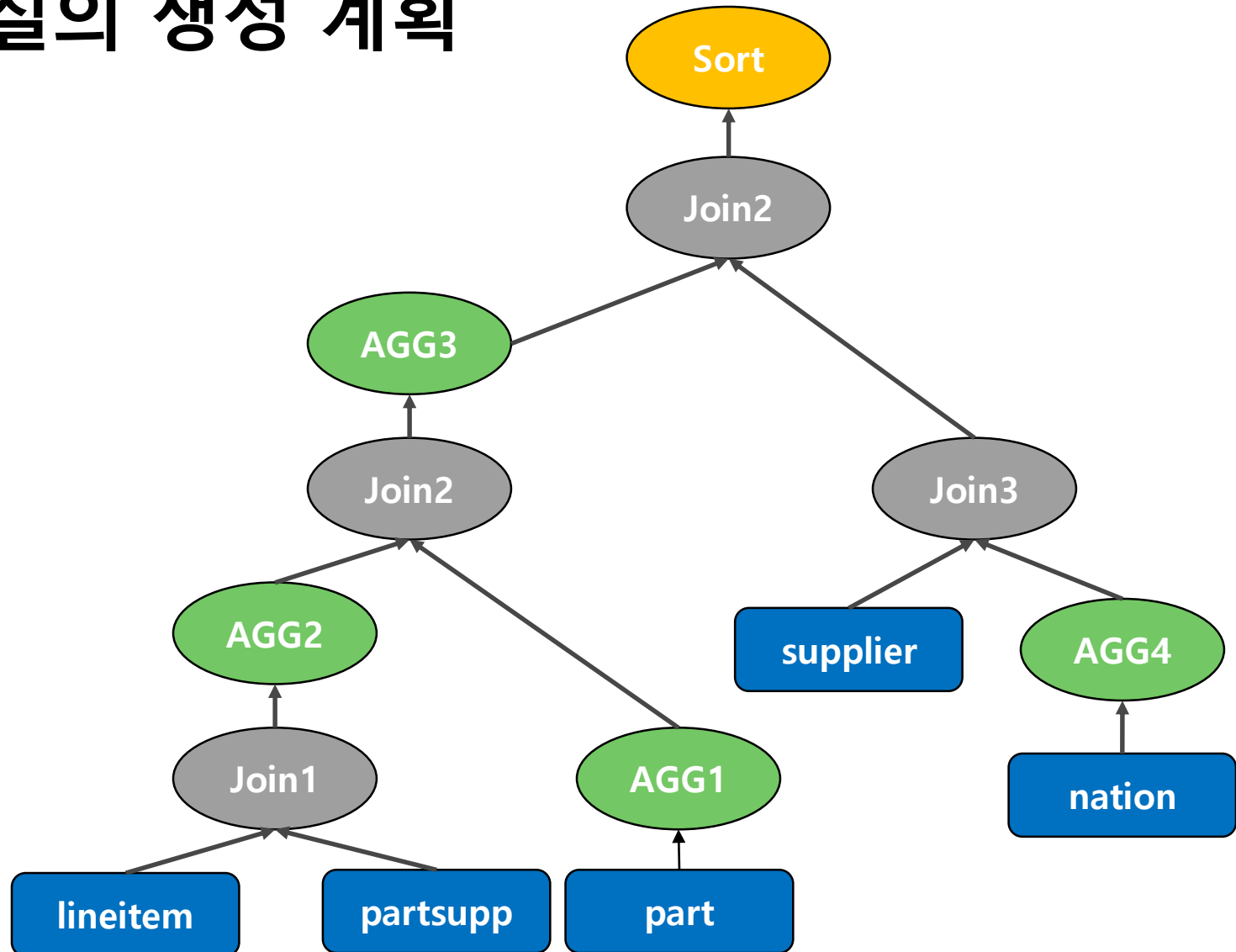
■ Q20 관련 질의문

```
select s_name, s_address from supplier, nation where s_suppkey in(
  select ps_suppkey from partsupp where ps_partkey in(
    select p_partkey from part where p_name like '[COLOR]%'
  )
  and ps_availqty > (
    select 0.5 * sum (l_quantity) from lineitem where
      l_partkey = ps_partkey
      and l_suppkey = ps_suppkey
      and l_shipdate >= date ('[DATE]')
      and l_shipdate < date ('[DATE]') + interval '1' year
  )
)
and s_nationkey = n_nationkey
and n_name = '[NATION]'
order by
s_name;
```

- COLOR: P_NAME 생성을 위해 정의된 값 목록 내에서 무작위로 선택됨
- DATE: 1993~1997년 사이 무작위로 선택된 연도의 1월 1일임
- NATION: N_NAME을 정의한 값 목록 내에서 무작위로 선택됨

Q20 in TCP-H (계속)

■ 질의 생성 계획



Q20 in TCP-H (계속)

- Q20 작업을 5개의 부분 작업(Sub Jobs)로 분할
 - 각 부분 작업은 하나의 MapReduce 작업으로 구현 가능
 - Job1

```
select
    0.5 * sum (l_quantity)
from
    lineitem
where
    l_partkey = ps_partkey and l_suppkey = ps_suppkey
    and l_shipdate >= date ('[DATE]')
    and l_shipdate < date ('[DATE]') + interval '1' year
```

- Job2

```
select
    p_partkey
from
    part
where
    p_name like '[COLOR]%'
```


Q20 in TCP-H (계속)

■ Job3

```
select
    ps_suppkey
from
    partsupp
where
    result of first job
    and ps_avaiqty > result of second job
```

■ Job4

```
select
    s_name, s_address
from
    partsupp
where
    result of third job
    and s_nationkey = n_nationkey
    and n_name = '[NATION]'
```

■ Job 5: Job4의 결과를 s_name 컬럼으로 정렬

Q20 in TCP-H (계속)

■ MapReduce Pseudo Code (Job1)

Class MAPPER_LINEITEM

method MAP (textline l , line L)

L : a recode of lineitem table

$d \leftarrow \text{'DATE'}$

$k \leftarrow \text{partkey} + \text{'|'} + \text{suppkey}$

$v \leftarrow \text{quantity}$

if ($\text{shipdate} \geq d$ and $\text{shipdate} < d + 1 \text{ year}$)

 EMIT (k , v)

Class MAPPER_PARTSUPP

method MAP (textline l , line L)

L : a recode of partsupp table

$k \leftarrow \text{partkey} + \text{'|'} + \text{suppkey}$

$v \leftarrow \text{availqty}$

EMIT (k , v)

Class REDUCER

method REDUCE (text k , valuse [v_1, v_2, \dots])

$L \leftarrow \text{new List}()$

if (availqty exist in valuse [v_1, v_2, \dots])

for all $v \in \text{valuse} [v_1, v_2, \dots]$ **do**

$\text{sum} \leftarrow \text{sum} + \text{quantity}$

$L \leftarrow k$

if ($\text{availqty} > 0.5 * \text{sum}$)

for all $k \in L$ **do**

$s[] \leftarrow \text{split}(k, \text{'|'})$

 EMIT ($s[0]$, $s[1]$)

Q20 in TCP-H (계속)

■ MapReduce Pseudo Code (Job2)

Class MAPPER_PART

method MAP (textline l , line L)

L : a recode of part table

$c \leftarrow \text{'COLOR'}$

$k \leftarrow \text{partkey}$

if ($\text{partname} == c$)

EMIT (k , null)

Class MAPPER_FIRST_RESULT

method MAP (textline l , line L)

L : a recode of first job result table

$k \leftarrow \text{partkey suppkey}$

$v \leftarrow \text{suppkey}$

EMIT (k , v)

Class REDUCER

method REDUCE (text k , valuse [v_1, v_2, \dots])

$L \leftarrow \text{new List()}$

for all $v \in \text{valuse } [v_1, v_2, \dots]$ **do**

if ($\text{values} == \text{null}$)

$p \leftarrow \text{TURE}$

else

$L \leftarrow v$

if ($p == \text{true}$)

for all $v \in L$ **do**

EMIT (v , null)

Q20 in TCP-H (계속)

■ MapReduce Pseudo Code (Job3)

Class MAPPER_SUPPLIER

method MAP (textline l , line L)

L : a recode of supplier table

$k \leftarrow s_supkey$

$v \leftarrow s_name \mid s_address \mid s_nationkey$

EMIT (k , v)

Class MAPPER_SECOND_RESULT

method MAP (textline l , line L)

L : a recode of second job result table

$k \leftarrow sec_result_supkey$

EMIT (k , $null$)

Class REDUCER

method REDUCE (text k , valuse [v_1, v_2, \dots])

$L \leftarrow \text{new List}()$

for all $v \in \text{valuse} [v_1, v_2, \dots]$ **do**

if ($\text{values} == null$)

$is_ps_supkey \leftarrow TURE$

else

$L \leftarrow v$

if (is_ps_supkey)

for all $v \in L$ **do**

EMIT (v , $null$)

Q20 in TCP-H (계속)

■ MapReduce Pseudo Code (Job4)

Class MAPPER_NATION

method MAP (textline l , line L)

L : a recode of nation table

$n \leftarrow \text{'NATION'}$

$k \leftarrow n_nationkey$

if ($n_nationname == n$)

EMIT (k , null)

Class MAPPER_THRID_RESULT

method MAP (textline l , line L)

L : a recode of third job result table

$k \leftarrow s_nationkey$

$v \leftarrow s_name + s_address$

EMIT (k , v)

Class REDUCER

method REDUCE (text k , valuse [v_1, v_2, \dots])

$L \leftarrow \text{new List}()$

for all $v \in \text{valuse } [v_1, v_2, \dots]$ **do**

if ($\text{values} == \text{null}$)

$\text{is_n_nationkey} \leftarrow \text{TURE}$

else

$L \leftarrow v$

if (is_n_nationkey)

for all $v \in L$ **do**

EMIT (v , null)

Q20 in TCP-H (계속)

■ MapReduce Pseudo Code (Job5)

```
Class MAPPER_SORT
```

```
method MAP (textline  $l$ , line  $L$ )
```

```
   $L$  : a recode of fourth job result table
```

```
   $k \leftarrow s\_name$ 
```

```
   $v \leftarrow s\_address$ 
```

```
  EMIT ( $k$ ,  $v$ )
```

```
Class REDUCER
```

```
method REDUCE (text  $k$ , valuse [ $v_1$ ,  $v_2$ , ...])
```

```
  EMIT ( $k$ ,  $v$ )
```

제출물

- TCP-H의 Q20중 Job1, Job2, Job3을 구현
- 아래 내용 제출
 - 소스코드 전문 + 예제 데이터
 - 예제 데이터는 구현하면서 사용한 작은 크기의 Sample Data
 - 구현 전반적인 설명, 실행화면 캡처화면, 실행 방법을 포함한 보고서 제출

MapReduce 알고리즘: Frequent Itemset Mining

담당교수: 전강욱(컴퓨터공학부)

kw.chon@koreatech.ac.kr

빈발 패턴 마이닝 (Frequent Itemset Mining)

- 데이터베이스에서 자주 나타나는 패턴을 모두 찾는 방법

Given a transaction database D and a minimum support $minsup$,
find all frequent itemsets that have the supports no less than $minsup$
- support: occurrence of an itemset in D

Input

Tid	Items
1	Beer, Nuts, Diaper
2	Beer, Coffee, Diaper
3	Beer, Diaper, Eggs
4	Nuts, Eggs, Milk
5	Coffee, Diaper, Milk

Finding the itemsets whose supports are no less than 3 in D

Output {Beer}, {Diaper}, and {Beer, diaper}

데이터마이닝 응용: Apriori(계속)

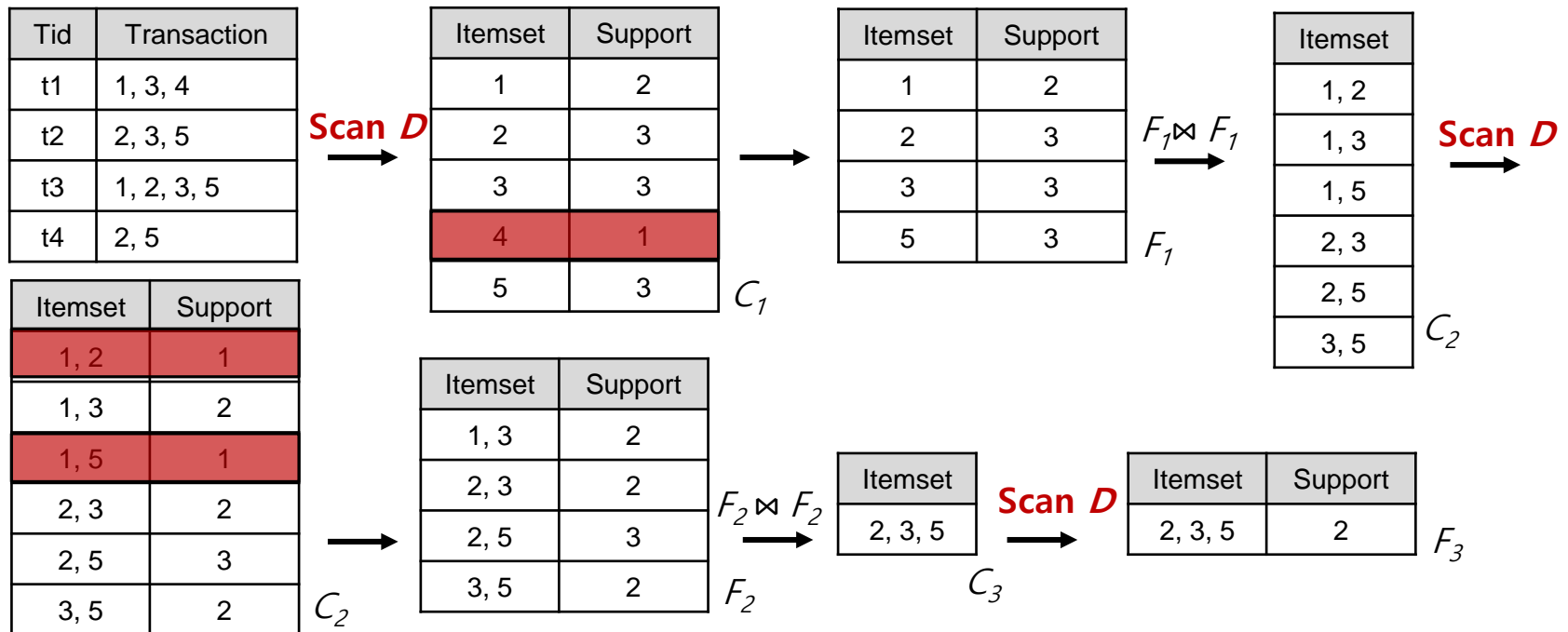
- 예제: 3번 이상 나타나는 모든 빈발 항목 집합을 찾아라

<i>D</i> =	1,2,5,6,7,9	included in at least 3 transactions
	2,3,4,5	
	1,2,7,8,9	
	1,7,9	
	2,7,9	
	2	
		{1} {2} {7} {9}
		{1,7} {1,9}
		{2,7} {2,9} {7,9}
		{1,7,9} {2,7,9}

데이터마이닝 응용: Apriori(계속)

■ Apriori 알고리즘

- 반복적인 후보집합생성 단계 및 테스트 단계



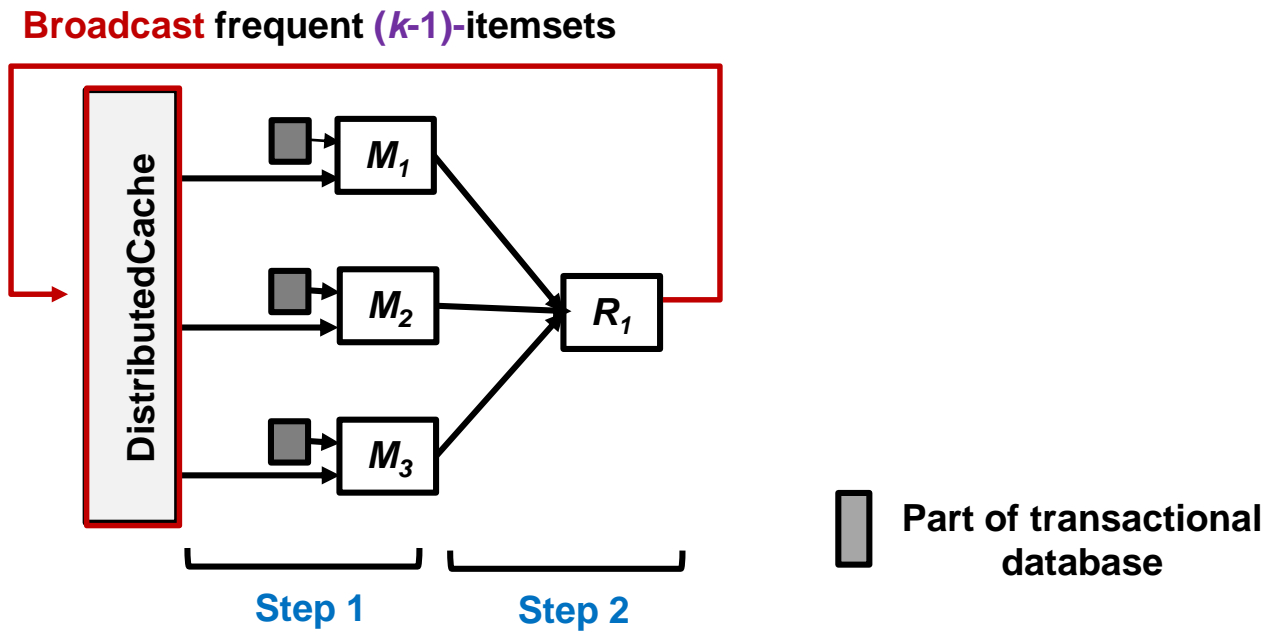
C_k : Set of candidate k -itemsets

F_k : Set of frequent k -itemsets

Example of Apriori when $minsup = 2$

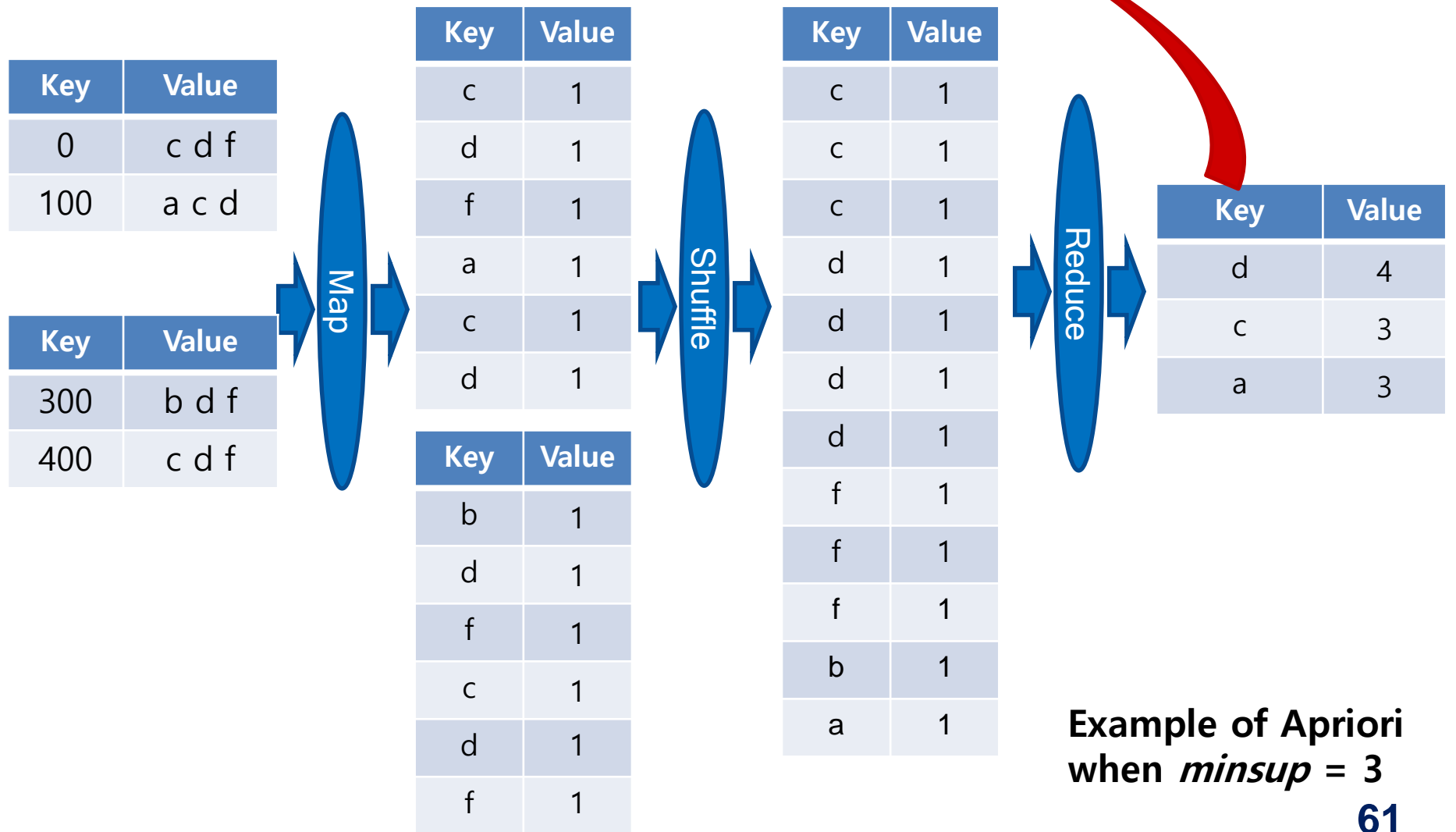
데이터마이닝 응용: Apriori(계속)

- MapReduce를 이용한 Apriori 알고리즘: 반복적인 후보집합생성 단계 및 테스트 단계(하나의 MapReduce 작업)를 더 이상 후보집합이 생성되지 않을 때까지 수행
 - Step1(K-번째 후보집합생성 단계): 후보 K-집합들을 생성하고 할당 받은 부분 데이터에 대해 후보 K-집합들의 부분 지지도를 계산
 - Step2(K-번째 테스트 단계): 모든 데이터에 대한 후보 K-집합들의 지지도를 계산하고, 빈발하지 않은 후보 K-집합 제거



데이터마이닝 응용: Apriori(계속)

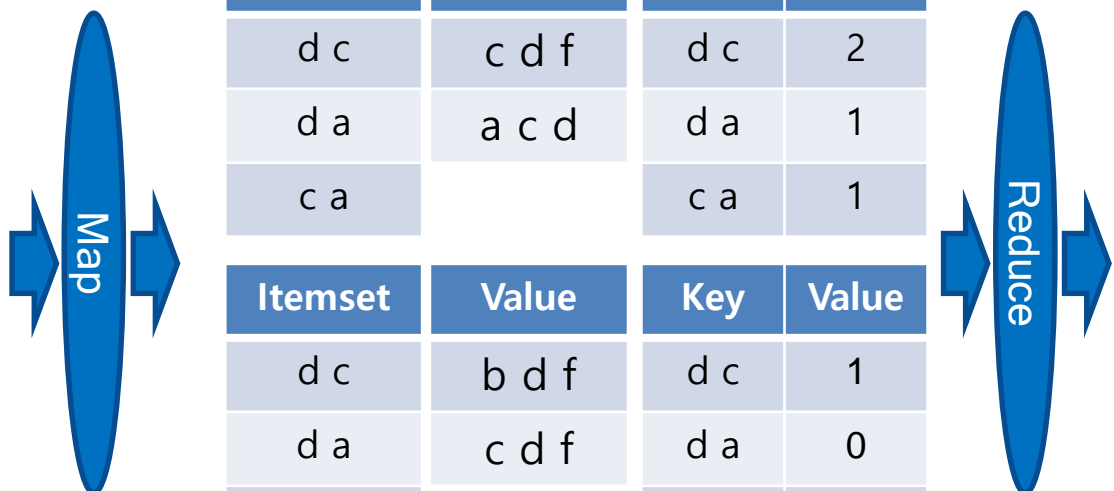
■ 첫 번째 단계 ($K=1$)



데이터마이닝 응용: Apriori(계속)

■ 이후 단계 ($K > 1$)

Check Candidates



Key	Value
0	c d f
100	a c d

Key	Value
300	b d f
400	c d f

Key	Value
d	4
c	3
a	3

Itemset	Value
d c	c d f
d a	a c d
c a	

Itemset	Value
d c	b d f
d a	c d f
c a	

Key	Value
d c	2
d a	1
c a	1

Key	Value
d c	1
d a	0
c a	0

Key	Value
d c	3

Output of the first iteration.

Broadcasted to each Mapper
Generate candidate 2-itemsets

Example of Apriori when $minsup = 3$

제출물

■ Apriori 알고리즘 구현

■ 아래 내용 제출

- 소스코드 전문 + 예제 데이터
 - 예제 데이터는 구현하면서 사용한 작은 크기의 Sample Data
- 구현 전반적인 설명, 실행화면 캡처화면, 실행 방법을 포함한 보고서 제출

MapReduce 알고리즘: K-Means Clustering

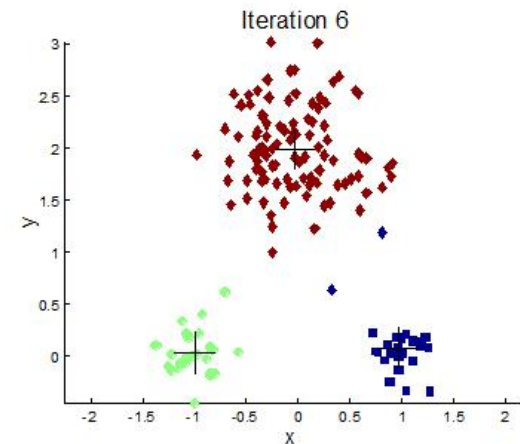
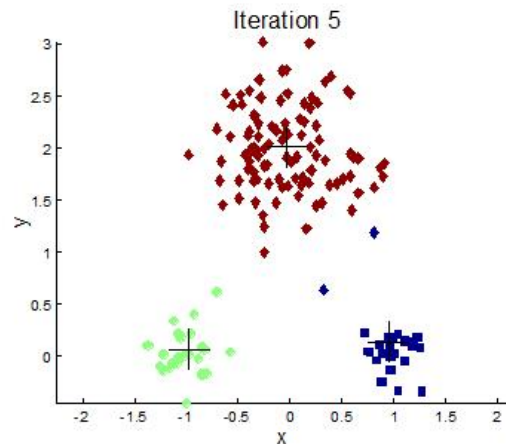
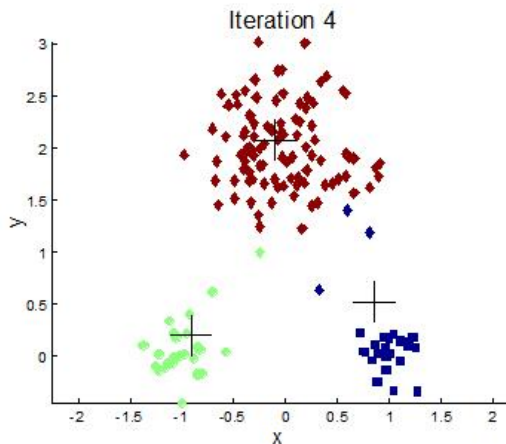
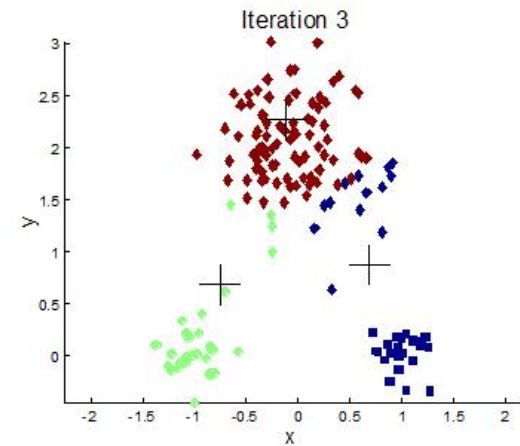
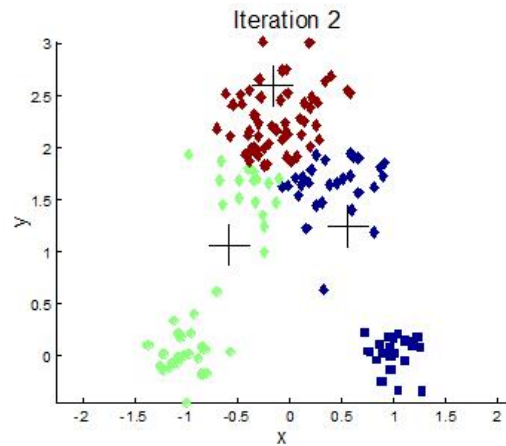
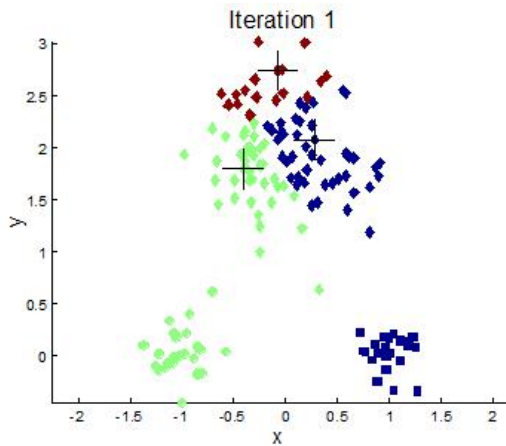
담당교수: 전강욱(컴퓨터공학부)

kw.chon@koreatech.ac.kr

데이터마이닝 응용: K-Means Clustering

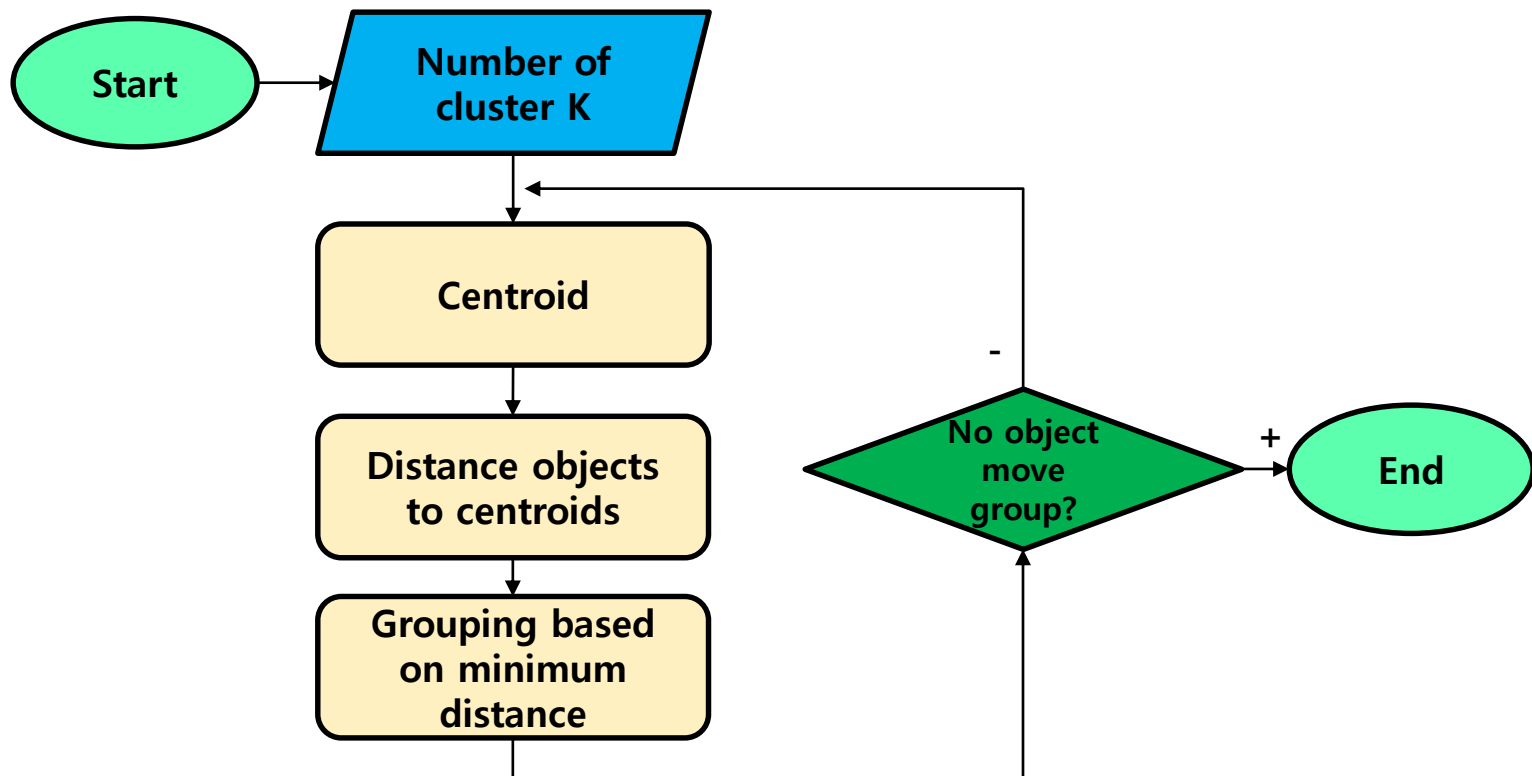
- **Clustering:** 특정 기준에 따라서 비슷한 데이터를 그룹으로 만드는 연산
 - 통계 분석, 이미지 분석, 정보 검색 등에서 사용
- **K-Means Clustering:** 데이터를 K개의 그룹으로 만드는 클러스터링
 - 데이터 내 각 포인트들과 Centroid 값과의 차이를 최소화 하는 방법으로 데이터를 그룹핑 함
 - Centroid: 클러스터의 중심부분에 위치한 (가상의) 포인트

데이터마이닝 응용: K-Means Clustering (계속)



데이터마이닝 응용: K-Means Clustering (예제)

■ 알고리즘 기술

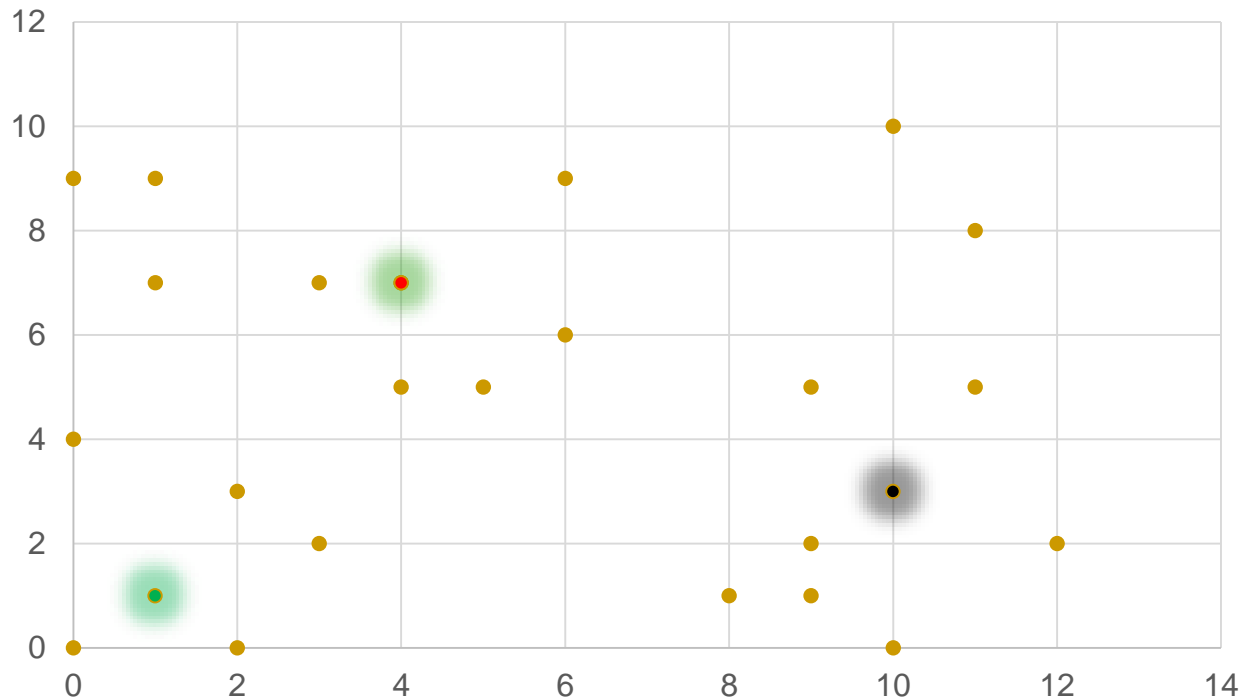


데이터마이닝 응용: K-Means Clustering (계속)

- MapReduce 기반 K-Means Clustering 방법은 Centroid 값이 변하지 않을 때까지 반복적(Iteratively)으로 동작함
 - Map: 각 Centroid 별로 가까운 포인트들을 모아서 Cluster를 생성함
 - Cluster 정보(포인트)는 입력 데이터임
 - Centroid 정보는 Distributed Cache를 통해서 모든 Mapper에 공유함
 - Cluster 정보와 Centroid 간 거리 값을 계산함
 - Cluster 정보는 가장 가까운 Centroid와 그룹핑 함
 - <Centroid, Cluster> 쌍이 중간 결과로 기록됨
 - Reduce: 새로운 Centroid 값을 계산
 - 같은 Key를 갖는 값들의 평균값을 계산함
 - 평균 값이 새로운 Centroid 값이 됨

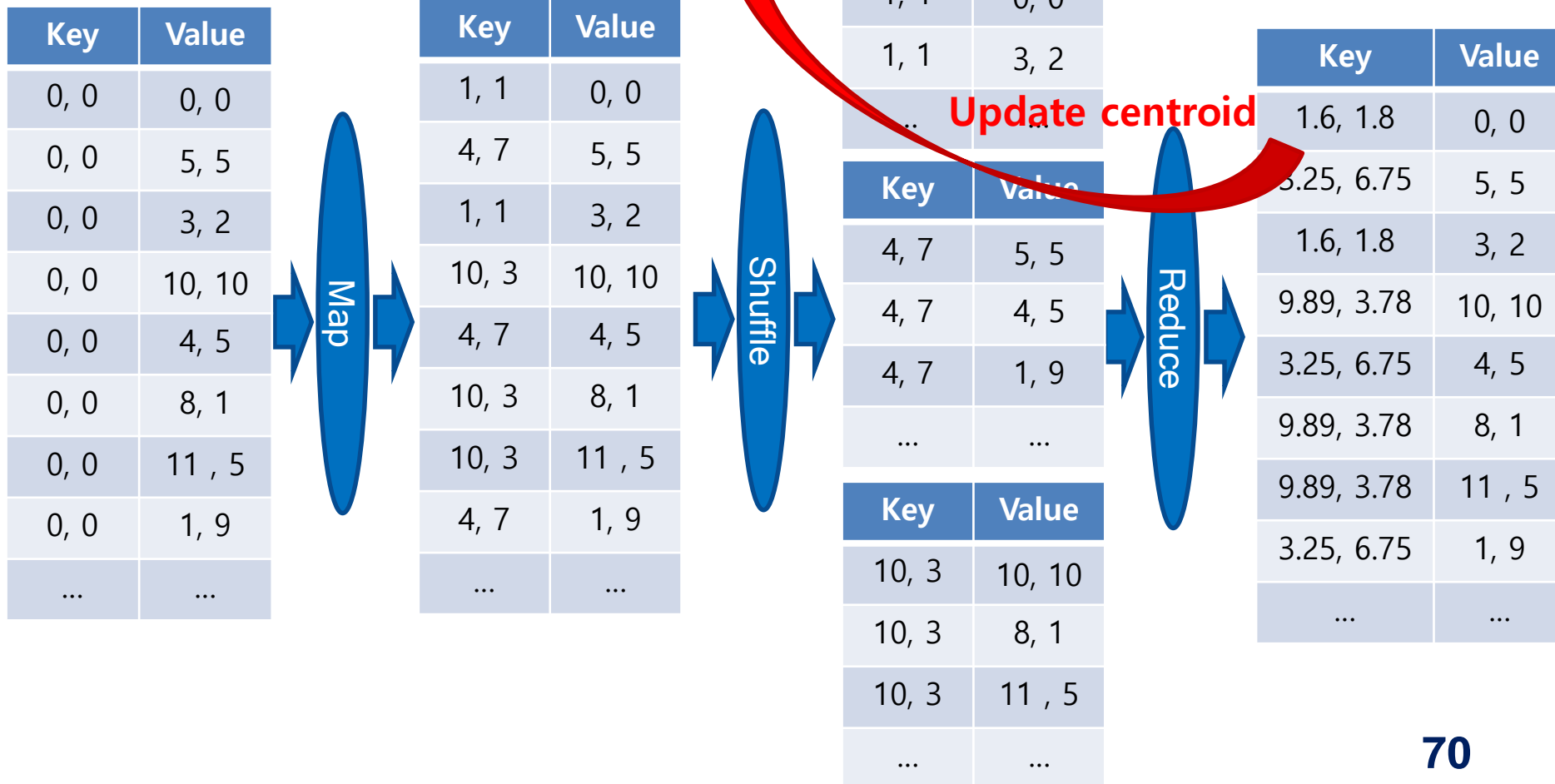
데이터마이닝 응용: K-Means Clustering (계속)

Centroid : (1, 1) (4, 7) (10, 3)



데이터마이닝 응용: K-Means Clustering (계속)

Centroid : (1, 1) (4, 7) (10, 3)



데이터마이닝 응용: K-Means Clustering (계속)

■ Pseudo Code

```
Class MAPPER
  method INITIALIZE
     $L \leftarrow \text{new List}()$ 
    Table  $T \leftarrow \text{DistributedCache}$ 
    for all centroid  $k \in T$  do
       $L \leftarrow k$ 
  method MAP (centroid  $cur\_k$ , clusters  $c$ )
    int  $d[]$ 
    for all centroid  $k \in L$  do
       $d[] \leftarrow \text{Distance}(k, c)$ 
     $cur\_k \leftarrow k$  of smallest distance
    EMIT (centroid  $cur\_k$ , clusters  $c$ )
```

데이터마이닝 응용: K-Means Clustering (계속)

■ Pseudo Code

```
Class REDUCER
     $L \leftarrow \text{new List}()$ 
    method REDUCE (centroid  $cur\_k$ , clusters [ $c_1, c_2, \dots$ ])
         $new\_centroid \leftarrow \text{mean}(\text{clusters}[c_1, c_2, \dots])$ 
         $L \leftarrow new\_centroid$ 
        EMIT (centroid  $cur\_k$ , clusters  $c$ )
    method CLEANUP
        for all centroid  $k \in L$  do
            DistributedCache  $\leftarrow k$ 
```


제출물

■ K-Means Clustering 알고리즘 구현

■ 아래 내용 제출

- 소스코드 전문 + 예제 데이터
 - 예제 데이터는 구현하면서 사용한 작은 크기의 Sample Data
- 구현 전반적인 설명, 실행화면 캡처화면, 실행 방법을 포함한 보고서 제출

감사합니다.

Contact: kw.chon@koreatech.ac.kr