

Spark R 소개

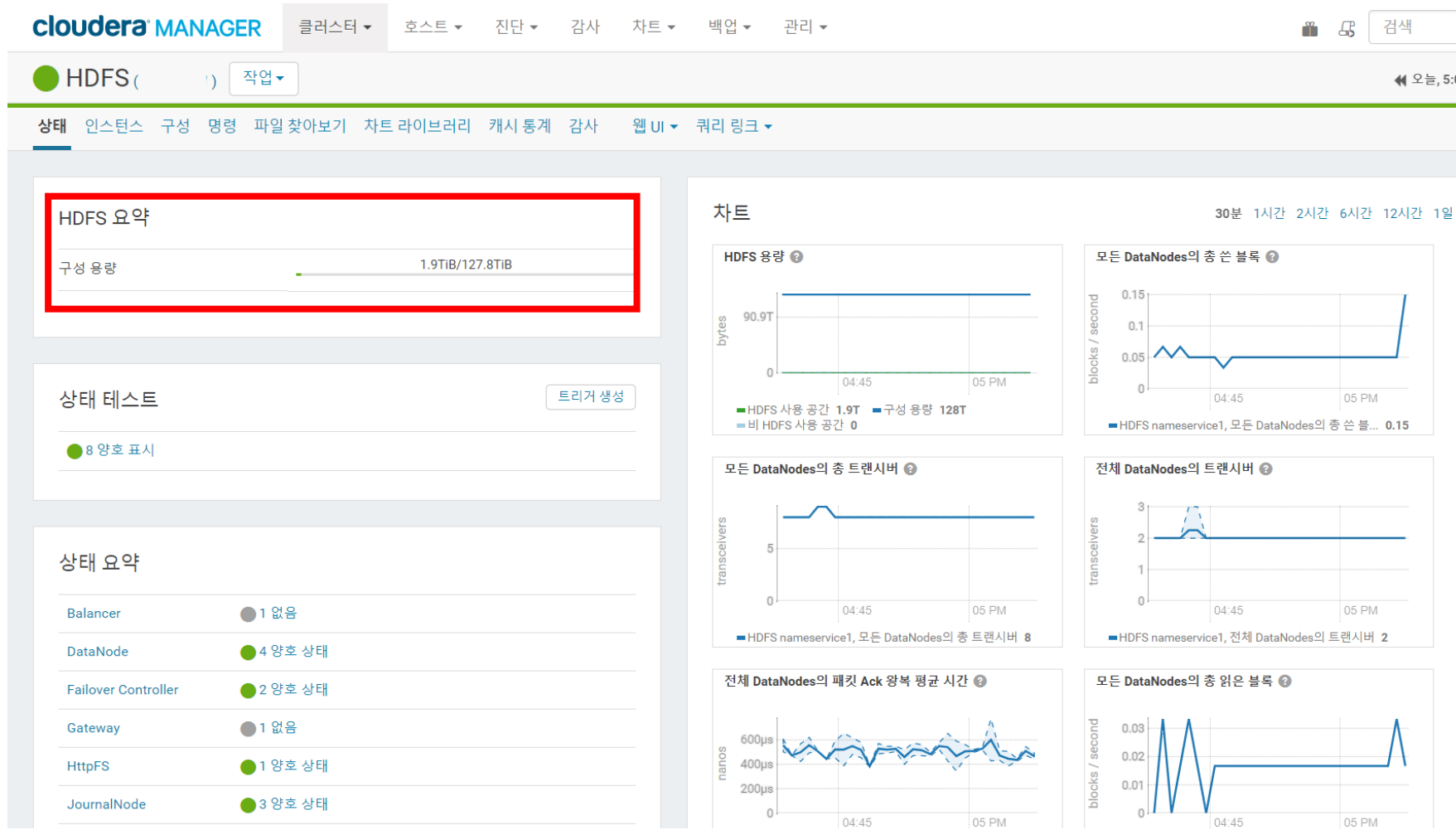
발표자 : 지용기

목차

1. 빅데이터 시스템이란?
2. Spark란 무엇인가?
3. Spark R 소개
4. Spark R 데모

1. 빅데이터 시스템이란?

- 거대한 데이터를 저장하고 그 데이터를 처리하기 위한 시스템



1. 빅데이터 시스템이란?

- 데이터를 조회하고 처리하기 위한 많은 오픈소스를 활용

The screenshot displays a web-based query editor interface. At the top, there is a navigation bar with tabs for 'Query Editors', 'Metastore Manager', 'Workflows', and 'Security'. A dropdown menu is open under 'Query Editors', listing 'Hive', 'Impala', 'DB 쿼리', 'Pig', and 'Job Designer'. The 'Hive' option is highlighted with a red box. In the main editor area, a SQL query is entered and highlighted with a red box:

```
1 select reg_date, count(*) from iot.sensor_data
2 where sensor_name like '%BER%'
3 group by reg_date ;
```

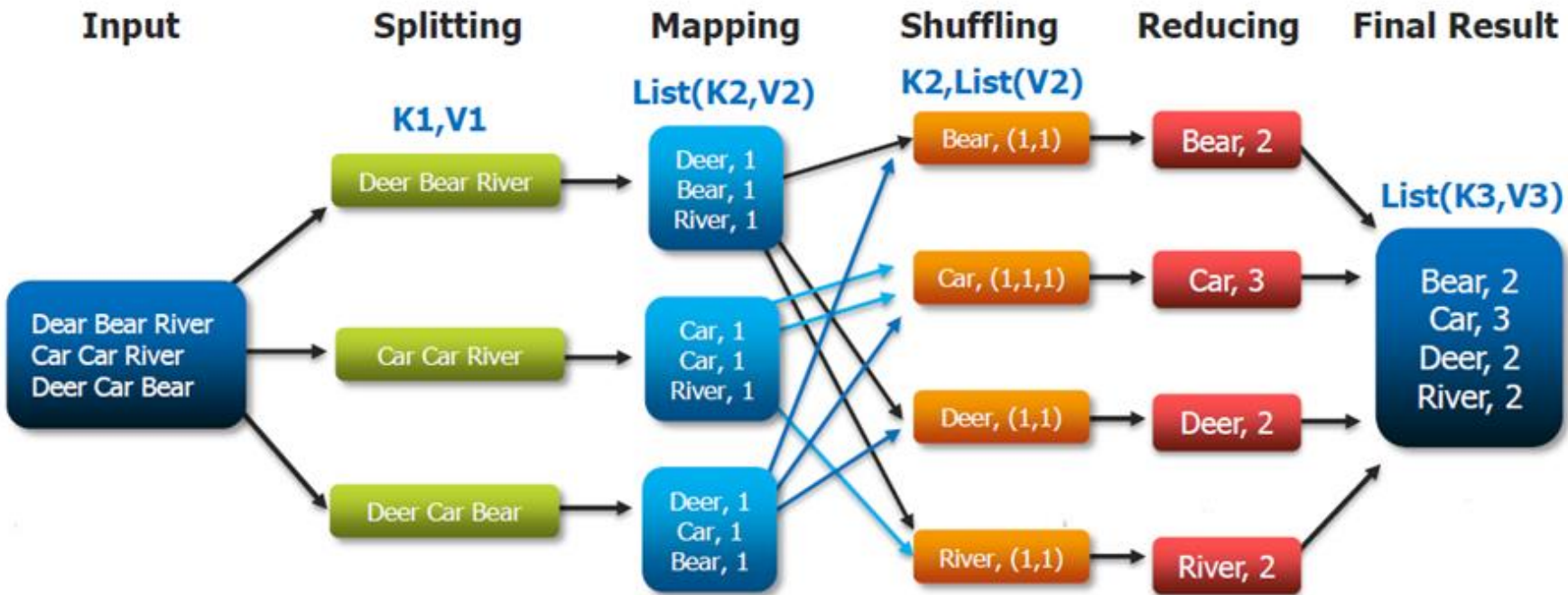
Below the query editor, there are tabs for '쿼리 기록', '저장된 쿼리', and '결과 (37)'. The '결과 (37)' tab is active, showing a table of query results. The table has two columns: 'reg_date' and 'count(*)'. The results are highlighted with a red box:

	reg_date	count(*)
1	20171121	79788864
2	20171213	81867961
3	20171207	80780007
4	20171208	81832629
5	20171117	71450478

1. 빅데이터 시스템이란?

- Mapreduce라는 병렬처리 프로그램 방식을 제공

The Overall MapReduce Word Count Process



```
import java.io.IOException;
import java.util.StringTokenizer;

import org.apache.hadoop.conf.Configuration;
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.output.FileOutputFormat;

public class WordCount {

    public static class TokenizerMapper
        extends Mapper<Object, Text, Text, IntWritable>{

        private final static IntWritable one = new IntWritable(1);
        private Text word = new Text();

        public void map(Object key, Text value, Context context
            ) throws IOException, InterruptedException {
            StringTokenizer itr = new StringTokenizer(value.toString());
            while (itr.hasMoreTokens()) {
                word.set(itr.nextToken());
                context.write(word, one);
            }
        }
    }

    public static class IntSumReducer
        extends Reducer<Text, IntWritable, Text, IntWritable> {
        private IntWritable result = new IntWritable();

        public void reduce(Text key, Iterable<IntWritable> values,
            Context context
            ) throws IOException, InterruptedException {

            int sum = 0;
            for (IntWritable val : values) {
                sum += val.get();
            }
            result.set(sum);
            context.write(key, result);
        }
    }

    public static void main(String[] args) throws Exception {
        Configuration conf = new Configuration();
        Job job = Job.getInstance(conf, "word count");
        job.setJarByClass(WordCount.class);
        job.setMapperClass(TokenizerMapper.class);
        job.setCombinerClass(IntSumReducer.class);
        job.setReducerClass(IntSumReducer.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);
        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));
        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
}
```

1. 빅데이터 시스템이란?

- 시연~~

2. Spark란 무엇인가?

- 지금까지 보여준 빅데이터 시스템들에는 문제점들이 존재함
 - ✓성능
 - ✓개발의 어려움
 - ✓기능의 제약

2. Spark란 무엇인가?

- 성능
 - ✓ 인메모리 : 데이터를 메모리에 올려놓고 처리하고 결과만 파일로 출력
- 개발의 어려움
 - ✓ 함수형 기반의 API 제공 : 함수형 프로그램 방식을 알면 쉬운데 모르면 더 어려움.
- 기능의 제약
 - SQL, 머신러닝, 실시간, 그래프 Library 제공

2. Spark란 무엇인가?

- 빅데이터를 처리하기 위한 병렬처리 프레임워크

The screenshot displays the Cloudera Manager interface for a Spark cluster. The top navigation bar includes '클러스터', '호스트', '진단', '감사', '차트', '백업', and '관리'. The main header shows 'Spark ()' with a '작업' dropdown. Below this, a secondary navigation bar contains '상태', '인스턴스', '구성', '명령', '차트 라이브러리', '감사', 'History Server Web UI', and '쿼리 링크'. The '상태' (Status) section is active, showing a '상태 테스트' button and a '트리거 생성' button. The '상태 요약' (Status Summary) table lists components and their health:

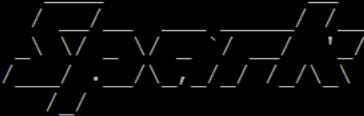
Component	Health
Gateway	1 없음 (No)
History Server	1 양호 상태 (Good)
호스트 (Host)	1 양호 상태 (Good)

The '차트' (Charts) section shows a '정보 이벤트' (Info Events) chart for the last 30 minutes, 1 hour, or 2 hours. The chart displays 0 events for 'Spark, 정보 이벤트'. Below it, the '중요 이벤트 및 알림' (Important Events and Alerts) section is also empty.

2. Spark란 무엇인가?

- Spark는 눈에 보이는 실체는 없음.(Library)
- 데이터를 처리하기 위한 명령어를 제공
- 지원하는 언어 : **scala**, java , python, R

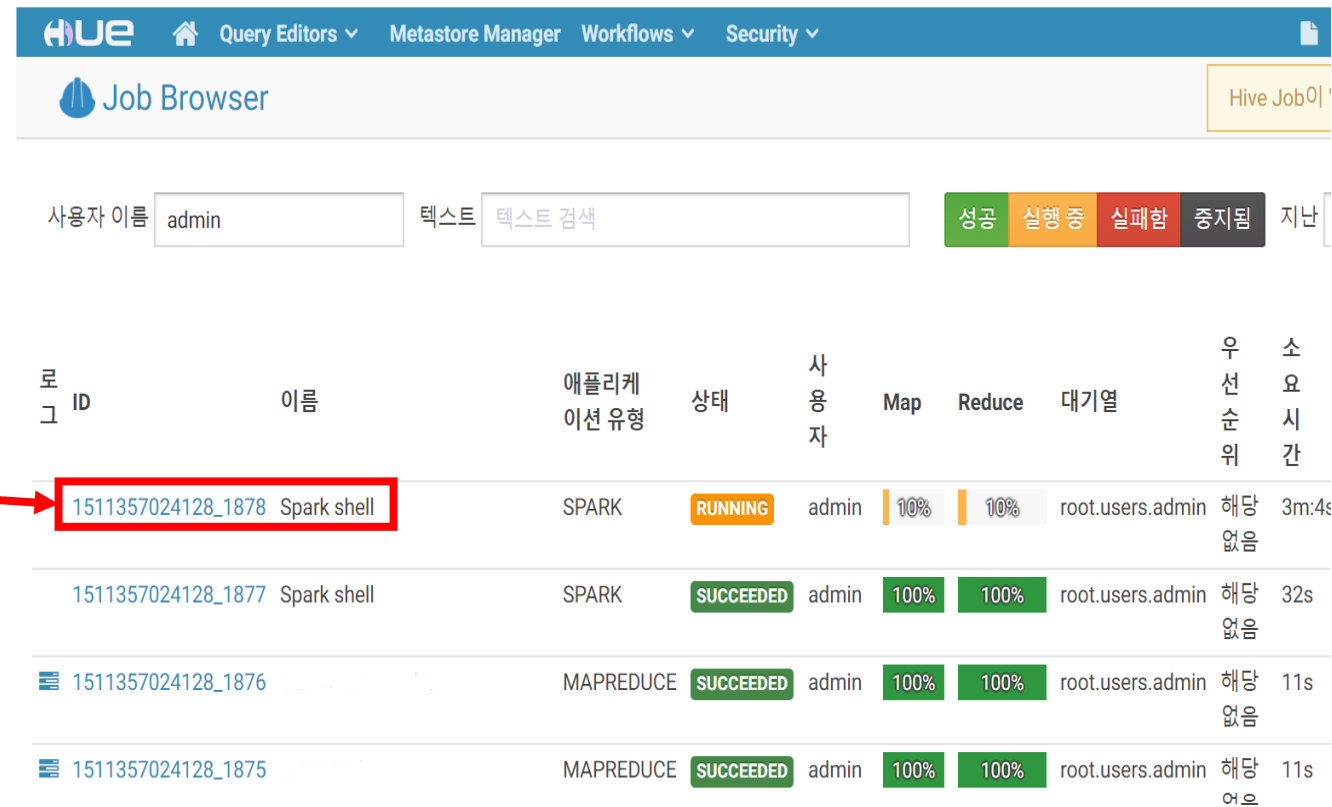
```
[admin@mgmt001 ~] $ spark-shell
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel).
Welcome to

 version 1.6.0

Using Scala version 2.10.5 (Java HotSpot(TM) 64-Bit Server VM, Java 1.7.0_67)
Type in expressions to have them evaluated.
Type :help for more information.
Spark context available as sc (master = yarn-client, app id = application_1511357024128_1878)
18/01/03 18:39:35 WARN metastore.ObjectStore: version information not found in metastore. hive.metastore.schema.verification is not enabled so recording the schema version 1.1.0
18/01/03 18:39:35 WARN metastore.ObjectStore: Failed to get database default, returning NoSuchObjectException
SQL context available as sqlContext.

scala> sc
res0: org.apache.spark.SparkContext = org.apache.spark.SparkContext@379d1c42

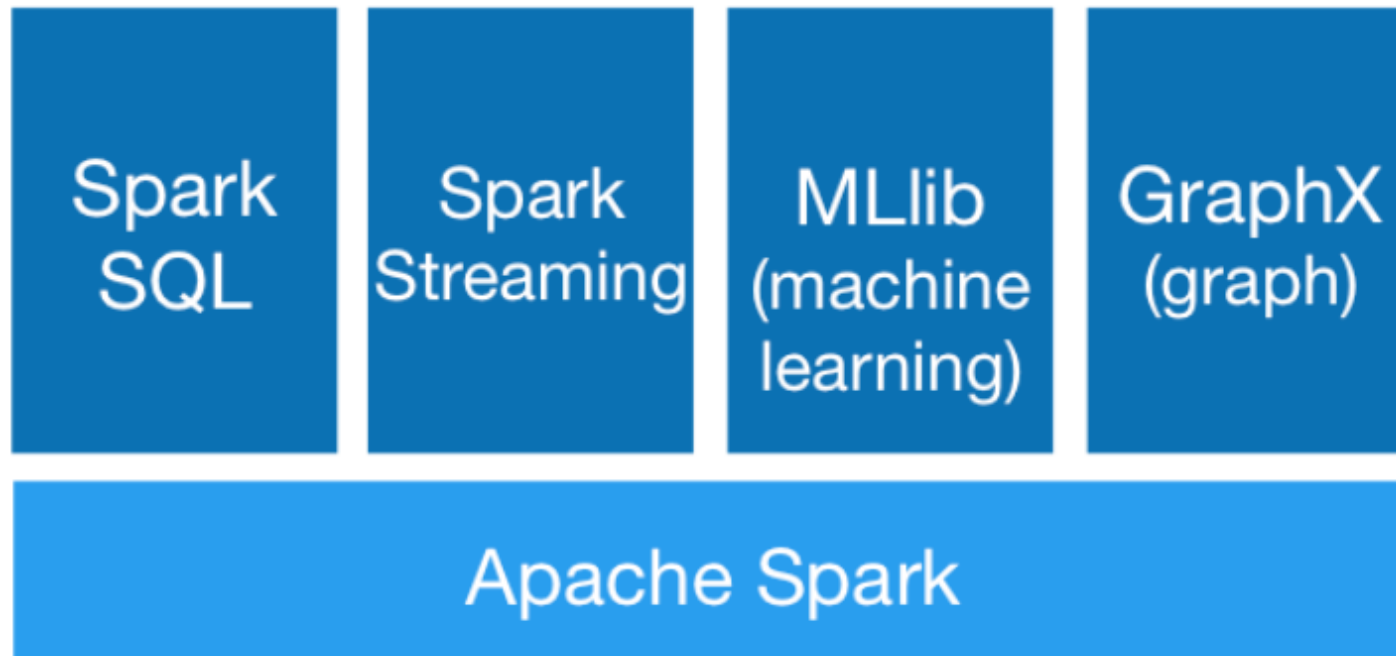
scala> █
```



로그 ID	이름	애플리케이션 유형	상태	사용자	Map	Reduce	대기열	우선 순위	소요 시간
1511357024128_1878	Spark shell	SPARK	RUNNING	admin	100%	100%	root.users.admin	해당 없음	3m:4s
1511357024128_1877	Spark shell	SPARK	SUCCEEDED	admin	100%	100%	root.users.admin	해당 없음	32s
1511357024128_1876	Spark shell	MAPREDUCE	SUCCEEDED	admin	100%	100%	root.users.admin	해당 없음	11s
1511357024128_1875	Spark shell	MAPREDUCE	SUCCEEDED	admin	100%	100%	root.users.admin	해당 없음	11s

2. Spark란 무엇인가?

- **Spark Core** : 데이터를 처리하기 위한 기본적인 명령어와 함수로 구성
- **Spark SQL** : 빅데이터를 SQL문으로 처리하기 위한 라이브러리
- Spark Streaming : 실시간 처리를 위한 라이브러리
- **MLlib** : 병렬처리를 지원하는 머신러닝 라이브러리
- GraphX : 병렬처리를 지원하는 graph 라이브러리



2. Spark란 무엇인가?

- Spark 시연 ~~
 - ✓ 윈도우에 가상머신 실행(Centos7.2)
 - ✓ Centos7.2에 docker 설치
 - ✓ [jupyter/all-spark-notebook](https://github.com/jupyter/all-spark-notebook) 라는 docker 이미지를 실행
- PySpark로 병렬처리 알고리즘 개발 예제
 - ✓ https://github.com/biospin/BigBio/blob/master/part05/week02_160705/data_algorithms/Chap14_NavieBayes.ipynb
 - ✓ https://github.com/biospin/BigBio/blob/master/part05/week04_160719/dataalgorith/Chap17_K-merCounting.ipynb
 - ✓ https://github.com/biospin/R_Bio/blob/master/part02/week2_160830/sparkR/Chap01_SecondarySort.ipynb

3. Spark R 소개

- SparkR은 R에서 Spark가 제공하는 Library를 사용하기 위한 R 패키지
- R의 dataframe과 비슷한 SparkDataFrame을 지원하고, MLlib도 지원

3. Spark R 소개

- SparkR에서 지원하지 않는 것들
 - ✓ 기존 R 패키지들이 SparkR에서 실행시키면 자동 병렬처리가 되지 않음.
 - ✓ SparkR의 Core API를 활용해서 새로운 병렬처리 머신러닝 알고리즘 개발이 거의 불가능

3. Spark R 소개

- SparkR에서 할 수 있는 것들
 - ✓ Spark에서 제공하는 API와 MLlib 을 활용
 - ✓ 빅데이터를 직접 R(Rstudio)에서 전처리하고 필터링해서 R의 풍부한 분석 알고리즘과 그래픽 함수로 처리 가능

3. Spark R 소개

- Spark + R 연동하는 Library가 2개 제공
 - ✓ Spark 진영 : <http://spark.apache.org/docs/latest/sparkr.html>
 - ✓ R 진영(sparklyr) : <http://spark.rstudio.com/>
- SparkR의 장점
 - ✓ 최신 Spark버전을 지원
- Sparklyr의 장단점
 - ✓ dplyr 형식의 데이터 처리 방식 지원
 - ✓ 최신 spark버전을 지원하지 않음
 - ✓ 아직 불안정함.

3. Spark R 소개

- Spark Core API만을 사용하는 SparkR 예제
- https://github.com/biospin/R_Bio/blob/master/part02/week2_160830/sparkR/SparkR_chap01.SecondarySort.ipynb
- https://github.com/biospin/R_Bio/blob/master/part02/week4_160920/SparkR_chap03.Top10List.ipynb

3. Spark R 소개

- Spark R 시연
 - ✓윈도우에 가상머신 실행(Centos7.2)
 - ✓Centos7.2에 docker 설치
 - ✓[jupyter/all-spark-notebook](https://hub.docker.com/r/jupyter/all-spark-notebook/) 라는 docker 이미지를 실행