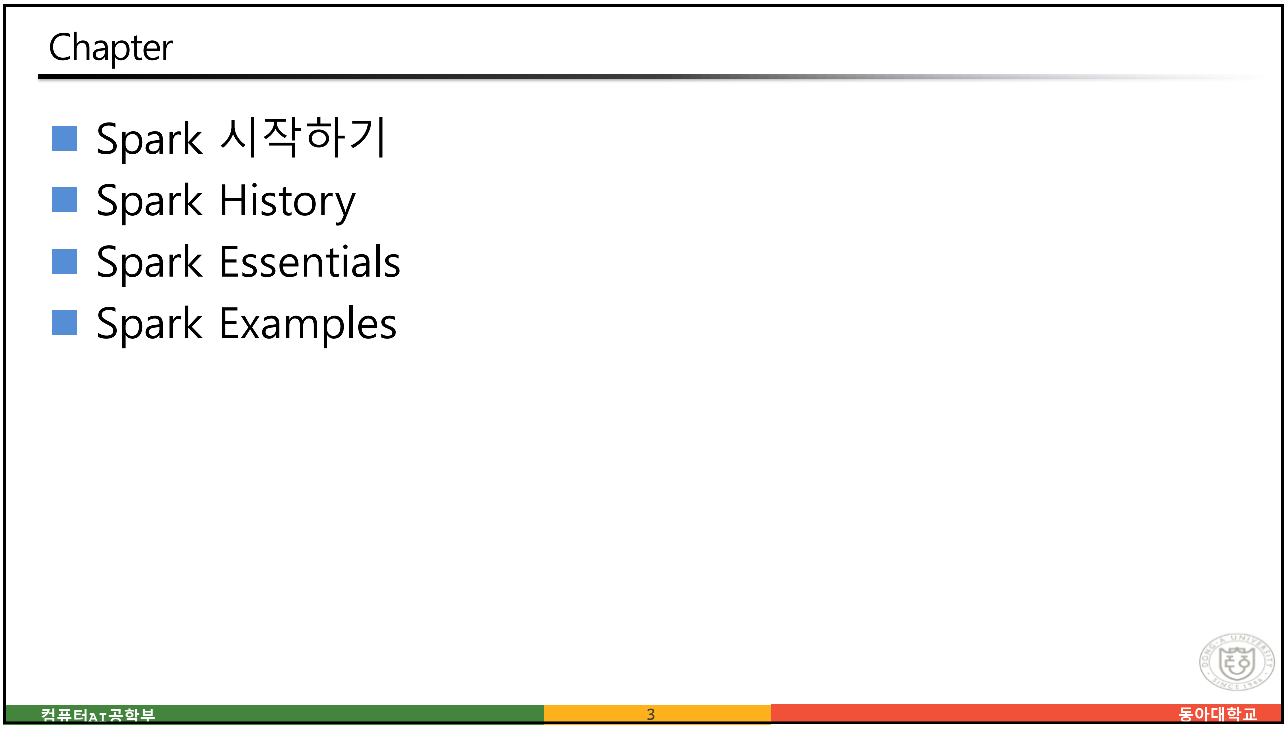
Apache Spark at-a-glance

빅데이터분석
천세진

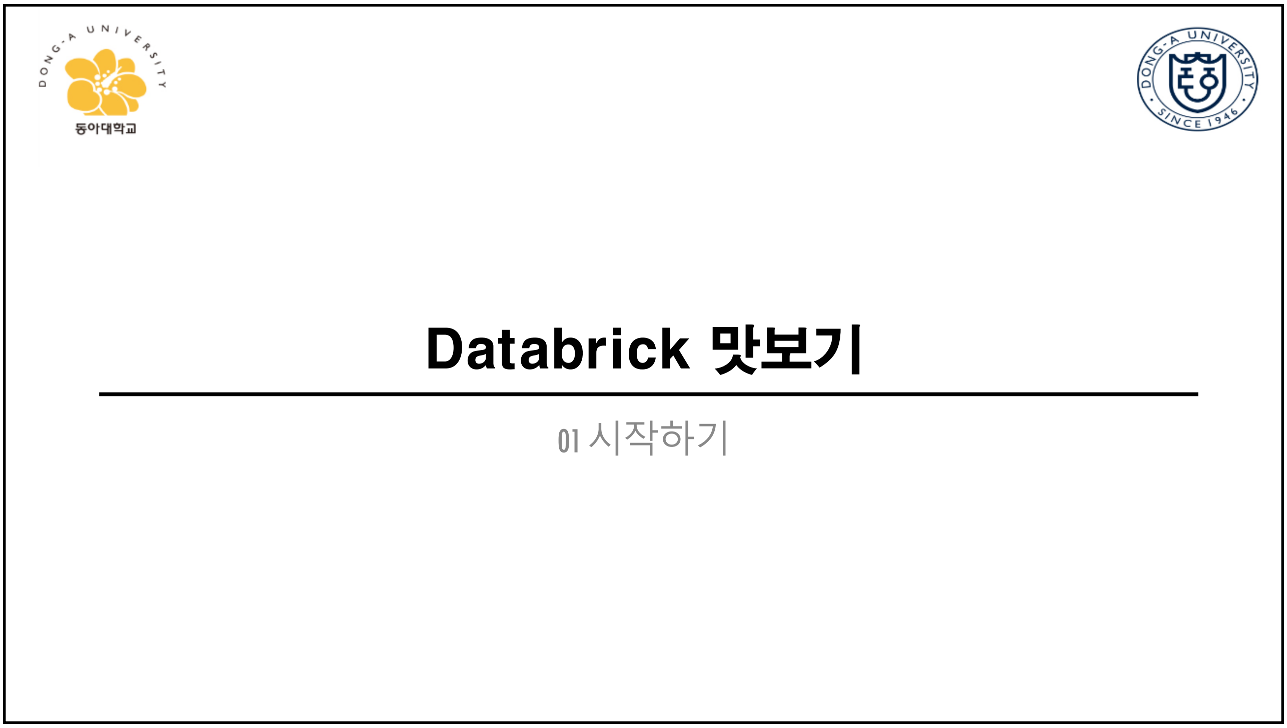
목표

- Spark shell 시작하기
- ML 알고리즘 사용하기
- HDFS로부터 데이터셋 탐색하기
- Spark SQL, Spark Streaming



Chapter

- Spark 시작하기
- Spark History
- Spark Essentials
- Spark Examples

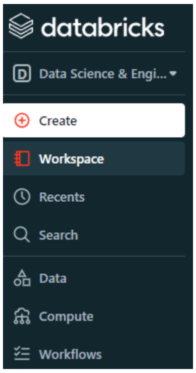


Databrick 맛보기

01 시작하기

Databricks 접속

- Cluster에 따라 Notebook 생성



RDD 생성해보기

- 데이터 생성하기

```
val data = 1 to 10000
```

RDD 생성해보기

- 데이터 생성하기

```
val data = 1 to 10000
```

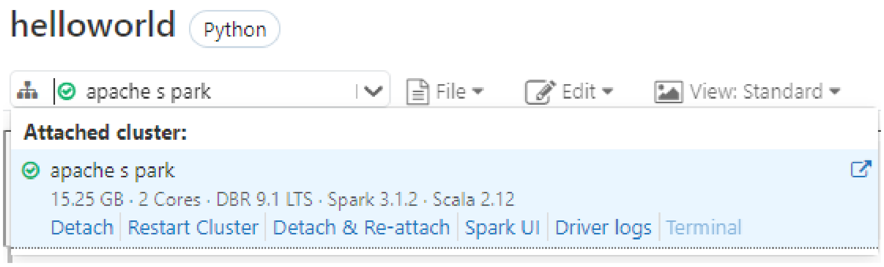
- RDD 기반 데이터를 생성하기

```
val distData = sc.parallelize(data)
```

- 10보다 이하 값에 대해서 필터 선택하기

```
distData.filter(_ < 10).collect()
```

Cluster 정보 확인



helloworld Python

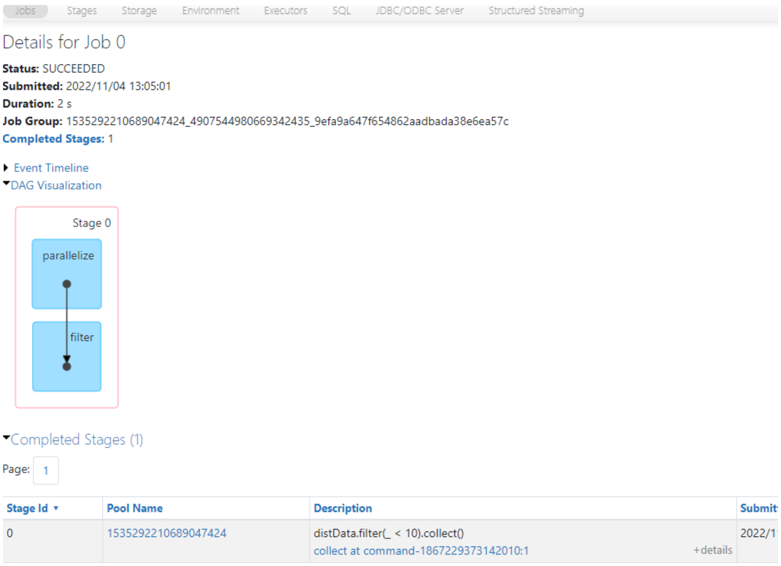
apache s park

Attached cluster:

apache s park

15.25 GB · 2 Cores · DBR 9.1 LTS · Spark 3.1.2 · Scala 2.12

Spark UI 확인



Details for Job 0

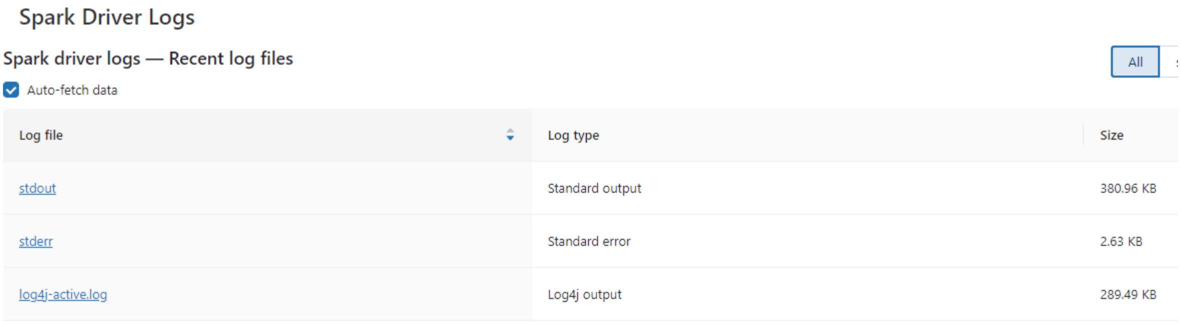
Status: SUCCEEDED
Submitted: 2022/11/04 13:05:01
Duration: 2 s
Job Group: 1535292210689047424_4907544980669342435_9efa9a647f654862aadbada38e6ea57c
Completed Stages: 1

▾Completed Stages (1)

Page: 1

Stage Id ▾	Pool Name	Description	Submit
0	1535292210689047424	distData.filter(_ < 10).collect() collect at command-1867229373142010:1 +details	2022/1

Driver logs



Spark Driver Logs

Spark driver logs — Recent log files

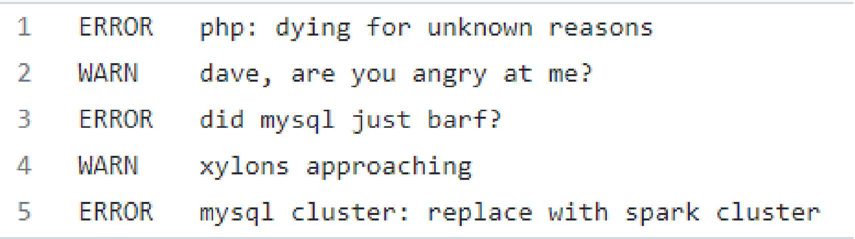
☑ Auto-fetch data

Log file	Log type	Size
stdout	Standard output	380.96 KB
stderr	Standard error	2.63 KB
log4j-active.log	Log4j output	289.49 KB

Spark Deconstructed

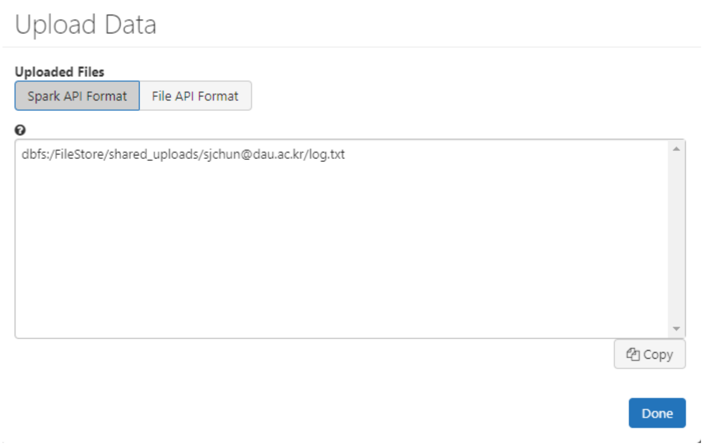
01 시작하기

Log file



```
ERROR   php: dying for unknown reasons
WARN    dave, are you angry at me?
ERROR   did mysql just barf?
WARN    xylons approaching
ERROR   mysql cluster: replace with spark cluster
```

Upload data



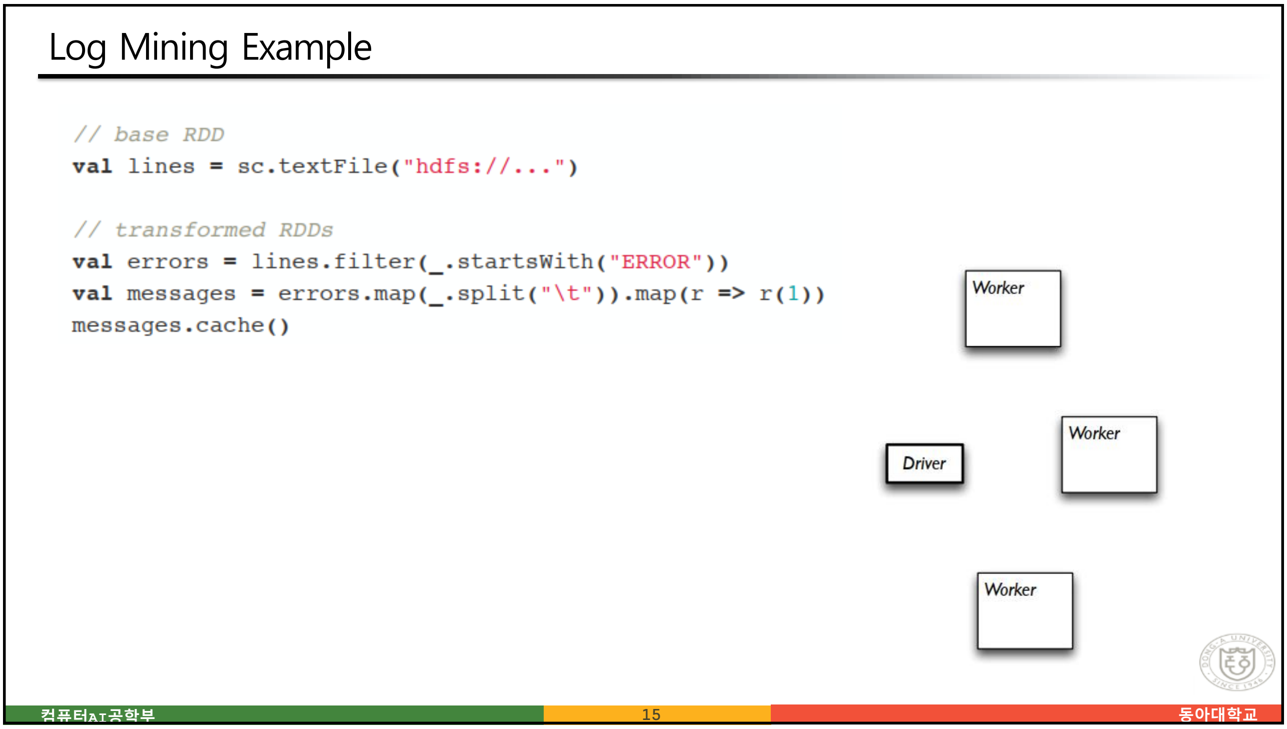
Upload Data

Uploaded Files

dbfs:/FileStore/shared_uploads/sjchun@dau.ac.kr/log.txt

Log Mining Example

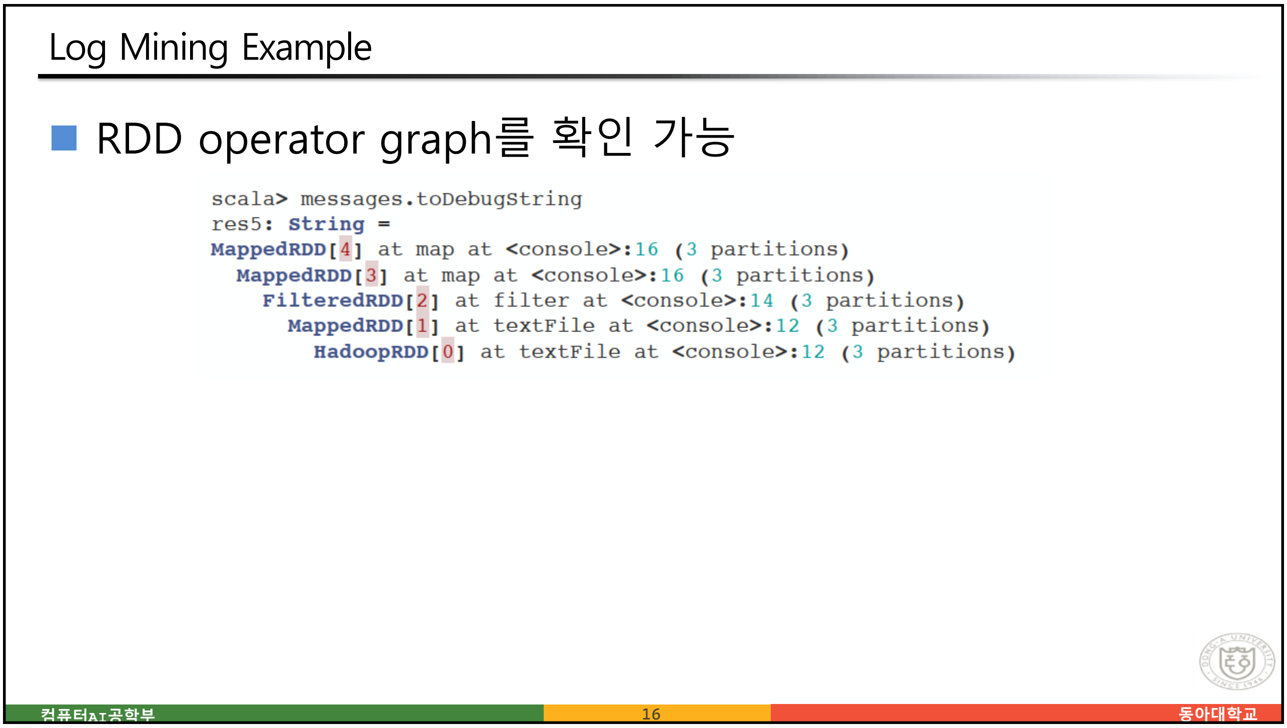
```
// base RDD
val lines = sc.textFile("hdfs://...")

// transformed RDDs
val errors = lines.filter(_.startsWith("ERROR"))
val messages = errors.map(_.split("\t")).map(r => r(1))
messages.cache()

// action 1
messages.filter(_.contains("mysql")).count()

// action 2
messages.filter(_.contains("php")).count()
```



Log Mining Example

```
// base RDD
val lines = sc.textFile("hdfs://...")

// transformed RDDs
val errors = lines.filter(_.startsWith("ERROR"))
val messages = errors.map(_.split("\t")).map(r => r(1))
messages.cache()
```

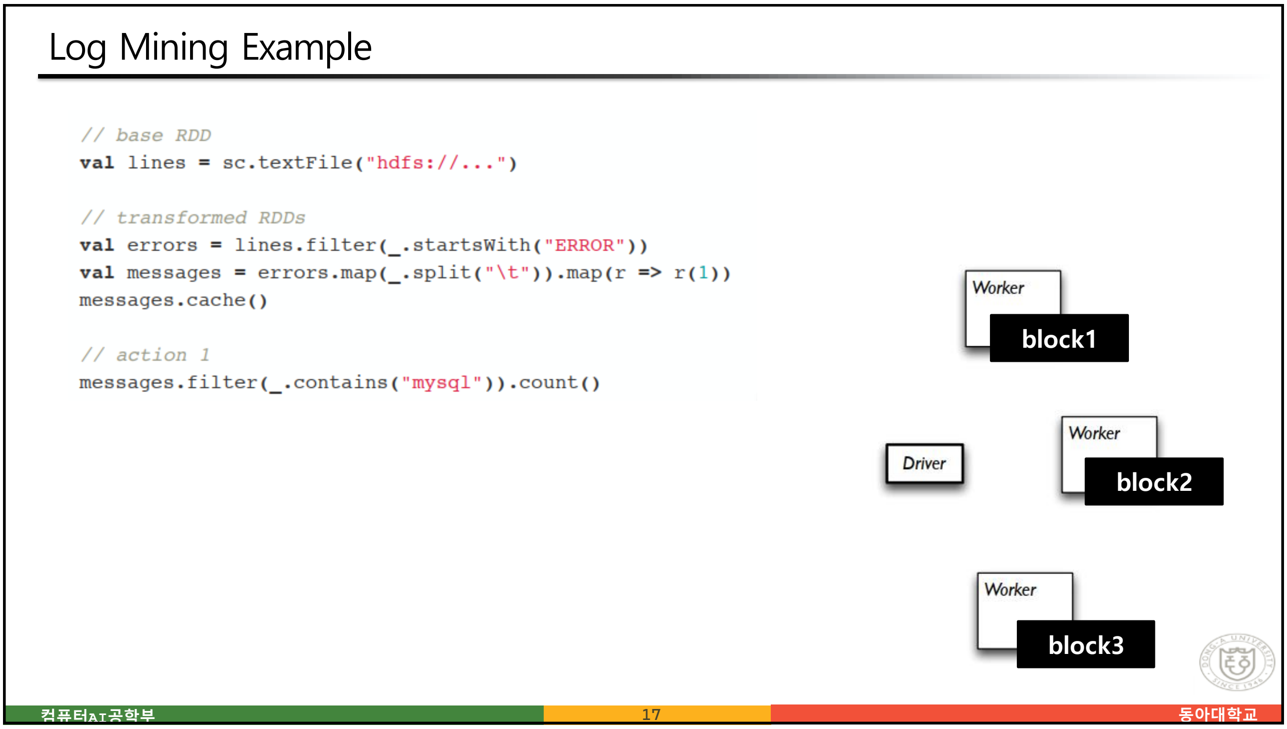
Worker
Driver
Worker
Worker



Log Mining Example

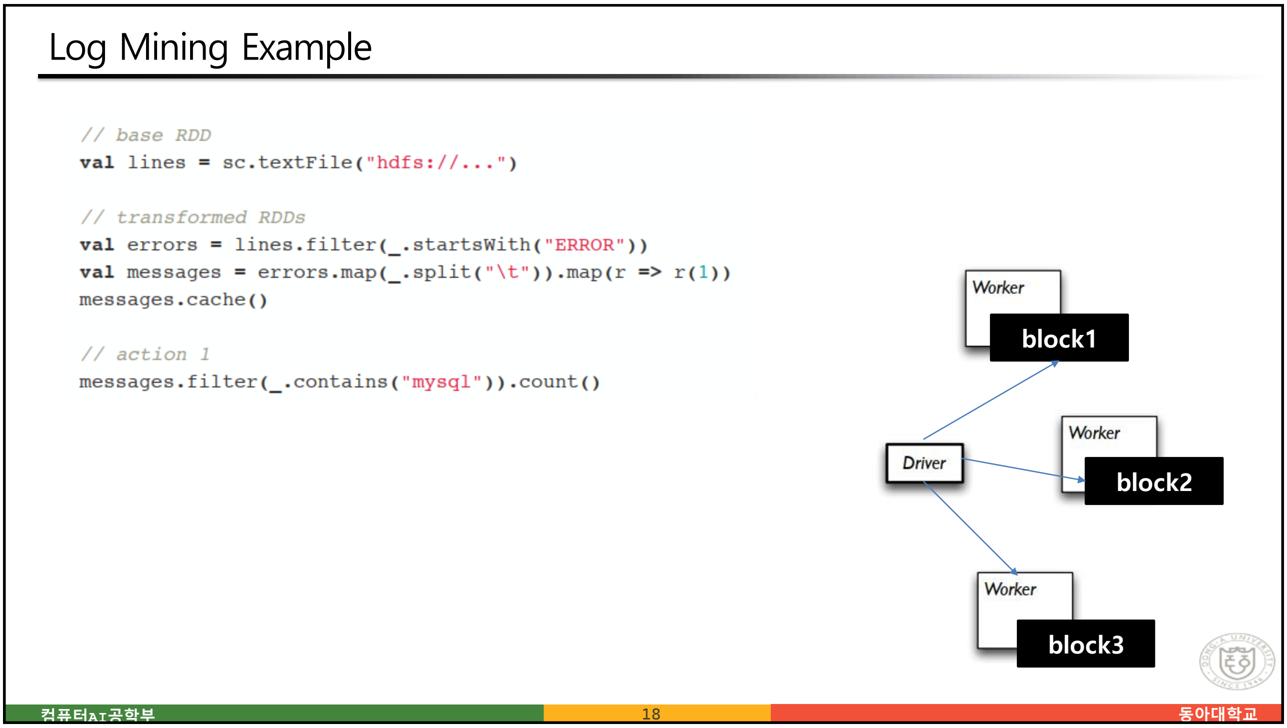
- RDD operator graph를 확인 가능

```
scala> messages.toDebugString
res5: String =
MappedRDD[4] at map at <console>:16 (3 partitions)
  MappedRDD[3] at map at <console>:16 (3 partitions)
    FilteredRDD[2] at filter at <console>:14 (3 partitions)
      MappedRDD[1] at textFile at <console>:12 (3 partitions)
        HadoopRDD[0] at textFile at <console>:12 (3 partitions)
```



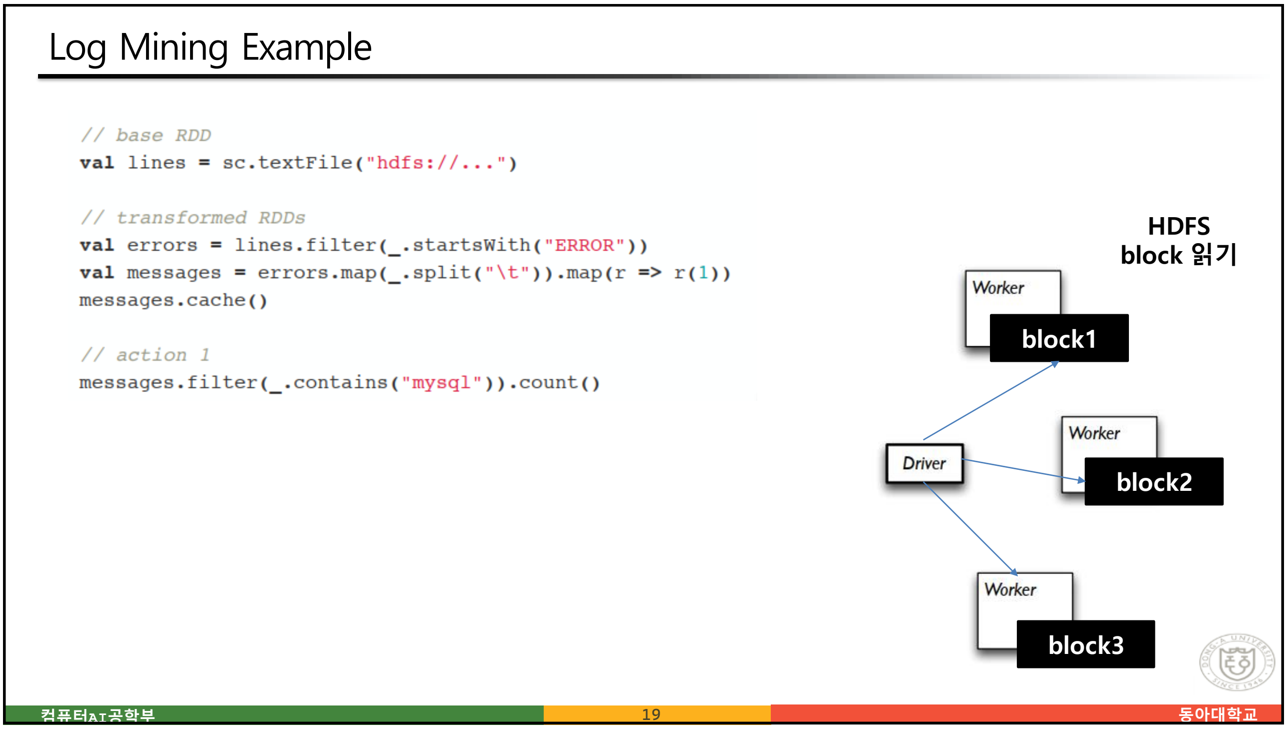
Log Mining Example

```
// base RDD
val lines = sc.textFile("hdfs://...")

// transformed RDDs
val errors = lines.filter(_.startsWith("ERROR"))
val messages = errors.map(_.split("\t")).map(r => r(1))
messages.cache()

// action 1
messages.filter(_.contains("mysql")).count()
```



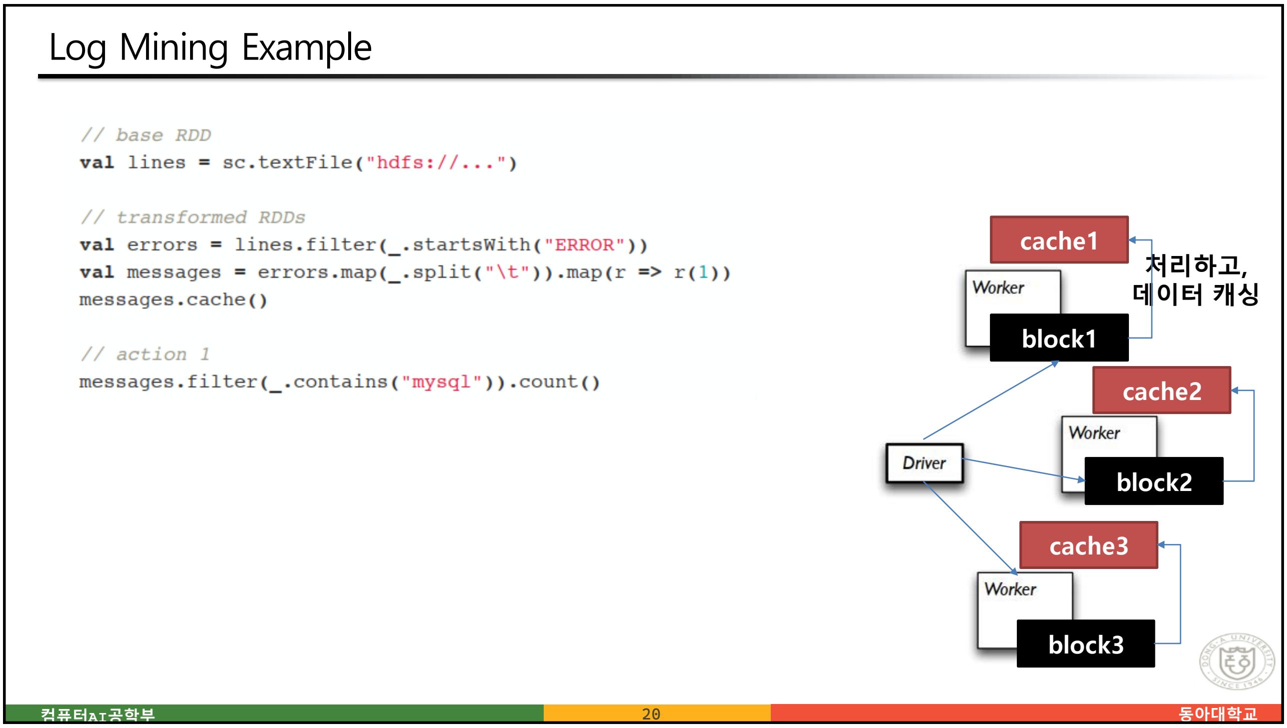
Log Mining Example

```
// base RDD
val lines = sc.textFile("hdfs://...")

// transformed RDDs
val errors = lines.filter(_.startsWith("ERROR"))
val messages = errors.map(_.split("\t")).map(r => r(1))
messages.cache()

// action 1
messages.filter(_.contains("mysql")).count()
```



Log Mining Example

```
// base RDD
val lines = sc.textFile("hdfs://...")

// transformed RDDs
val errors = lines.filter(_.startsWith("ERROR"))
val messages = errors.map(_.split("\t")).map(r => r(1))
messages.cache()

// action 1
messages.filter(_.contains("mysql")).count()
```

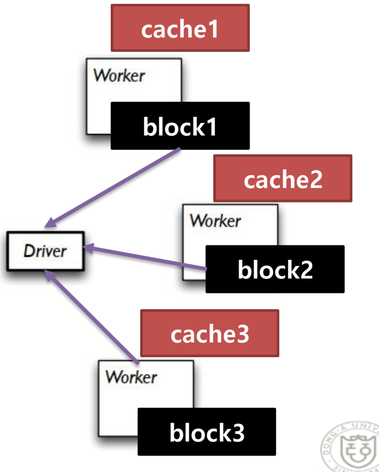


Log Mining Example

```
// base RDD
val lines = sc.textFile("hdfs://...")

// transformed RDDs
val errors = lines.filter(_.startsWith("ERROR"))
val messages = errors.map(_.split("\t")).map(r => r(1))
messages.cache()

// action 1
messages.filter(_.contains("mysql")).count()
```

Log Mining Example

```
// base RDD
val lines = sc.textFile("hdfs://...")

// transformed RDDs
val errors = lines.filter(_.startsWith("ERROR"))
val messages = errors.map(_.split("\t")).map(r => r(1))
messages.cache()

// action 1
messages.filter(_.contains("mysql")).count()
```

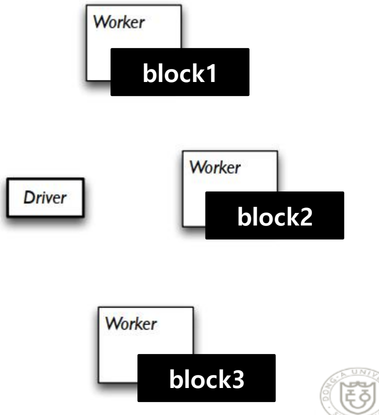


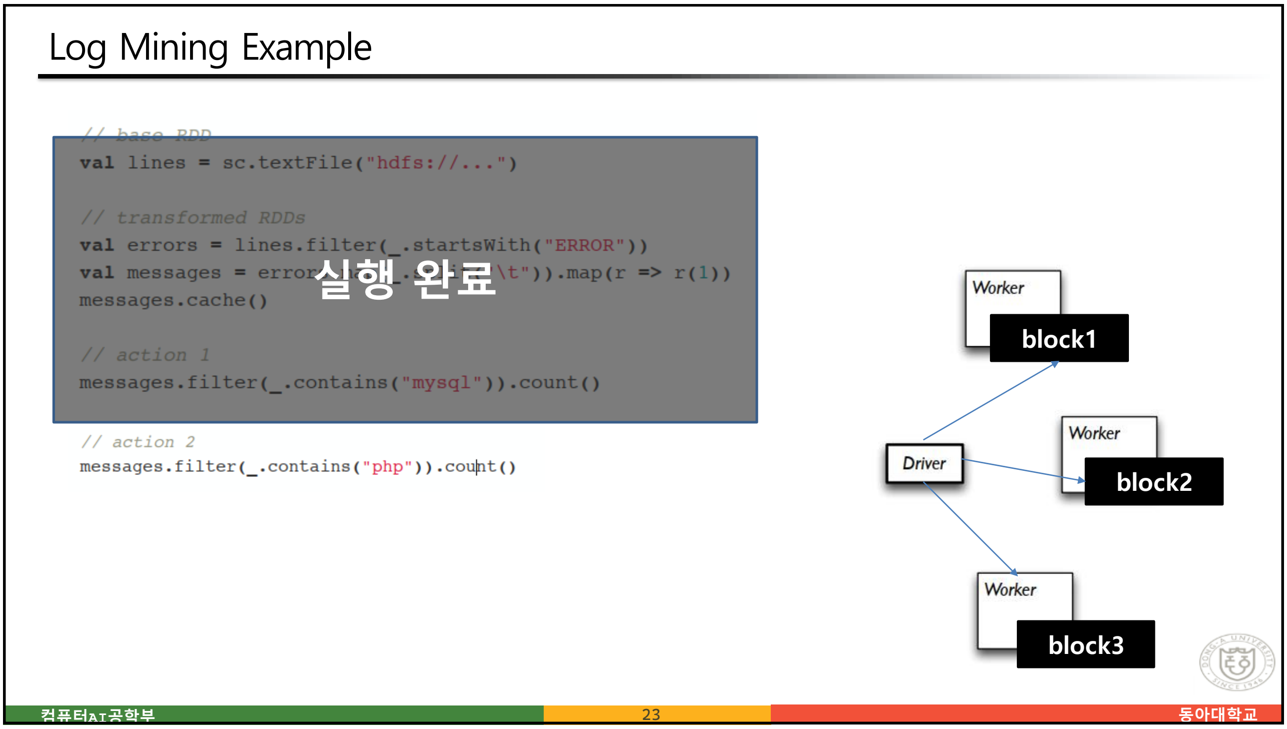
Log Mining Example

```
// base RDD
val lines = sc.textFile("hdfs://...")

// transformed RDDs
val errors = lines.filter(_.startsWith("ERROR"))
val messages = errors.map(_.split("\t")).map(r => r(1))
messages.cache()

// action 1
messages.filter(_.contains("mysql")).count()
```

실행 완료

```
// action 2
messages.filter(_.contains("php")).count()
```

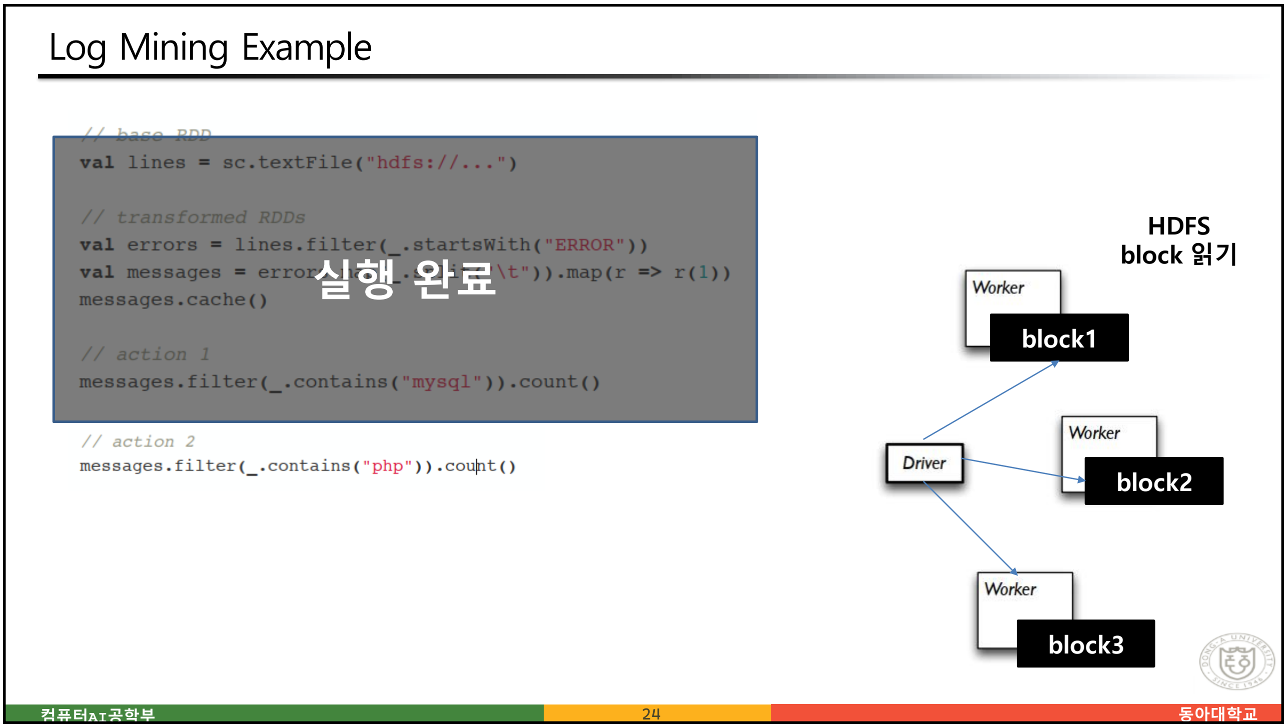




Log Mining Example

```
// base RDD
val lines = sc.textFile("hdfs://...")

// transformed RDDs
val errors = lines.filter(_.startsWith("ERROR"))
val messages = error            \t")).map(r => r(1))
messages.cache()

// action 1
messages.filter(_.contains("mysql")).count()

// action 2
messages.filter(_.contains("php")).count()
```

실행 완료



Log Mining Example

```
// base RDD
val lines = sc.textFile("hdfs://...")

// transformed RDDs
val errors = lines.filter(_.startsWith("ERROR"))
val messages = error            \t")).map(r => r(1))
messages.cache()

// action 1
messages.filter(_.contains("mysql")).count()

// action 2
messages.filter(_.contains("php")).count()
```

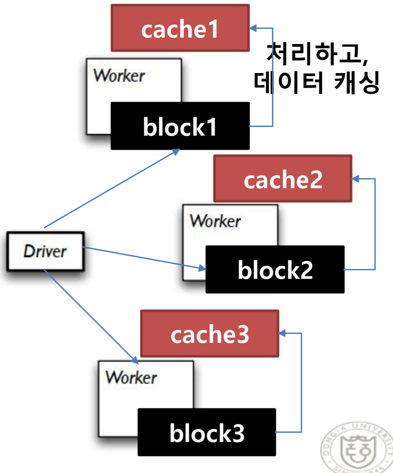
실행 완료

HDFS block 읽기

Log Mining Example

```
// base RDD
val lines = sc.textFile("hdfs://...")

// transformed RDDs
val errors = lines.filter(_.startsWith("ERROR"))
val messages = errors.map(_.split("\t")).map(r => r(1))
messages.cache()

// action 1
messages.filter(_.contains("mysql")).count()

// action 2
messages.filter(_.contains("php")).count()
```

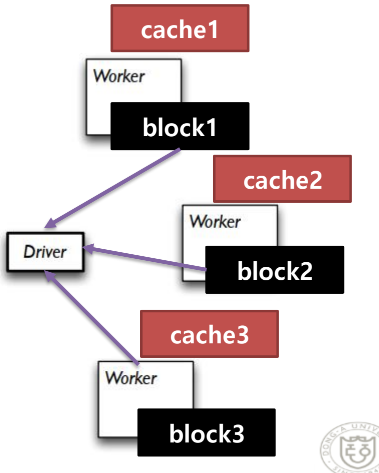
실행 완료



Log Mining Example

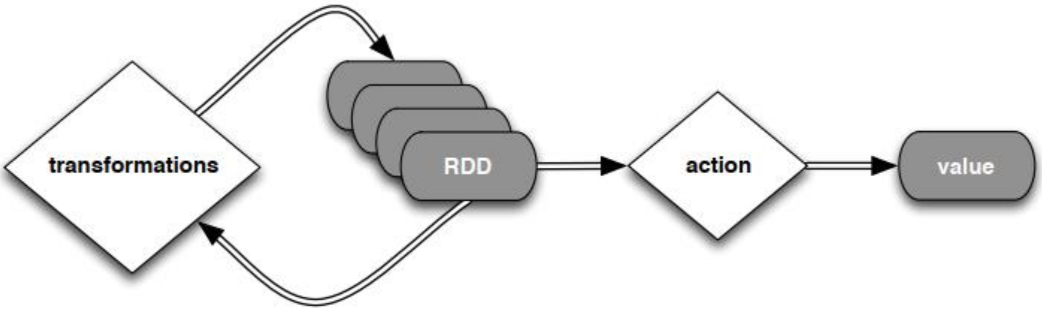
```
// base RDD
val lines = sc.textFile("hdfs://...")

// transformed RDDs
val errors = lines.filter(_.startsWith("ERROR"))
val messages = errors.map(_.split("\t")).map(r => r(1))
messages.cache()

// action 1
messages.filter(_.contains("mysql")).count()

// action 2
messages.filter(_.contains("php")).count()
```

실행 완료

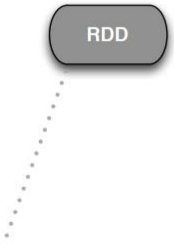


Spark Deconstructed

- RDD transformations and actions

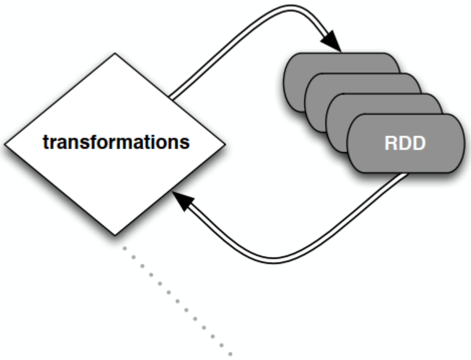


Spark Deconstructed



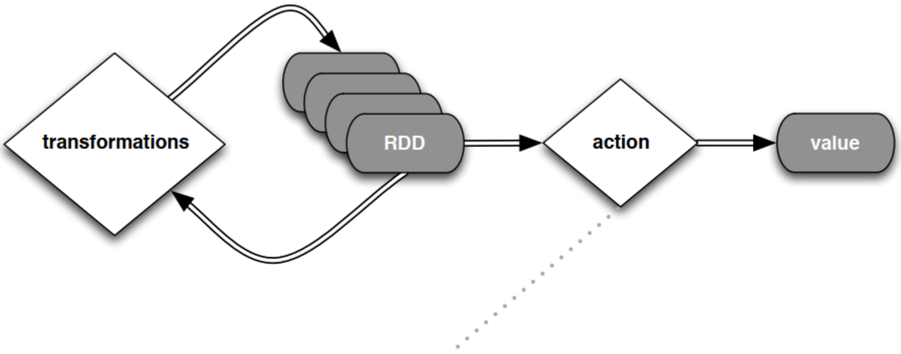
```
// base RDD
val lines = sc.textFile("hdfs://...")
```

Spark Deconstructed



```
// transformed RDDs
val errors = lines.filter(_.startsWith("ERROR"))
val messages = errors.map(_.split("\t")).map(r => r(1))
messages.cache()
```

Spark Deconstructed



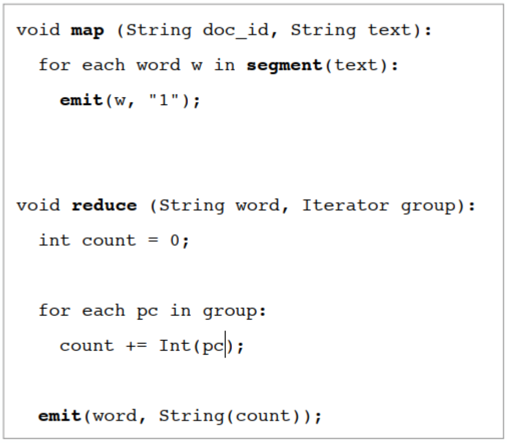
```
// action 1
messages.filter(_.contains("mysql")).count()
```

Simple Spark Apps

01 시작하기

Word Count

- Text 문서의 콜렉션 내에서, 각 단어들이 얼마나 나타나는지를 세기
- 병렬적으로 처리하는 방법



```
void map (String doc_id, String text):
  for each word w in segment(text):
    emit(w, "1");


void reduce (String word, Iterator group):
  int count = 0;

  for each pc in group:
    count += Int(pc);

  emit(word, String(count));
```

Word Count

Scala:

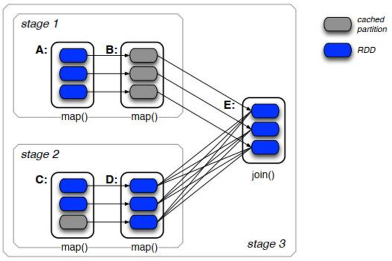
```
val f = sc.textFile("README.md")
val wc = f.flatMap(l => l.split(" ")).map(word => (word, 1)).reduceByKey(_ + _)
wc.saveAsTextFile("wc_out.txt")
```

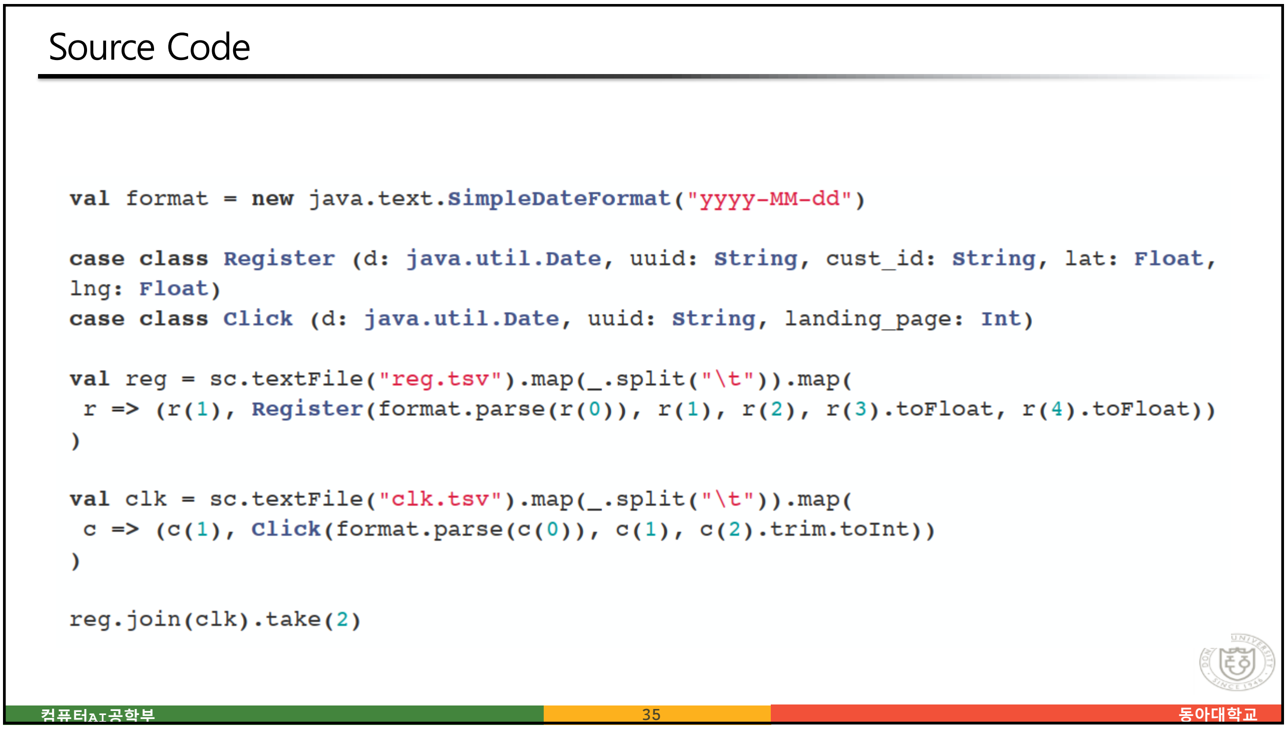
Python:

```
from operator import add
f = sc.textFile("README.md")
wc = f.flatMap(lambda x: x.split(' ')).map(lambda x: (x, 1)).reduceByKey(add)
wc.saveAsTextFile("wc_out.txt")
```

Word Count

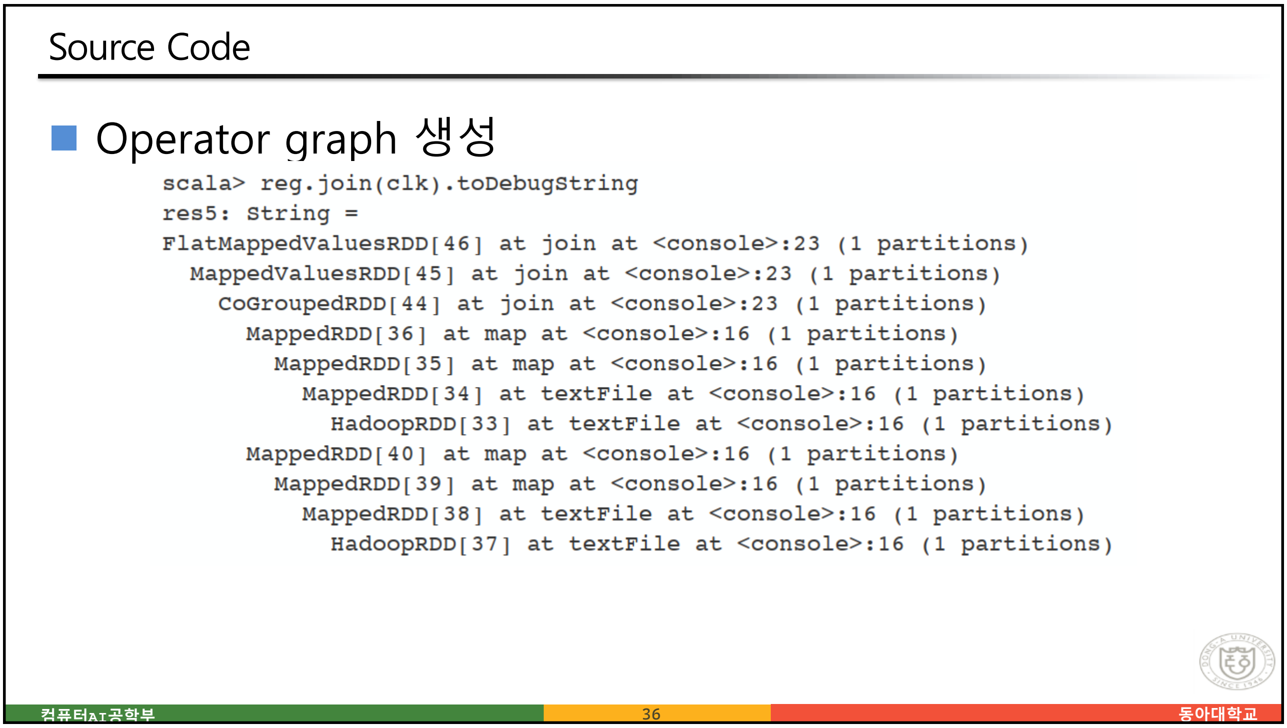
```
2014-03-04      15dfb8e6cc4111e3a5bb600308919594        11
2014-03-06      81da510acc4111e387f3600308919594        61

2014-03-02      15dfb8e6cc4111e3a5bb600308919594       1       33.6599436237    -117.958125229
2014-03-04      81da510acc4111e387f3600308919594       2       33.8570099635    -117.855744398
```





Source Code

```
val format = new java.text.SimpleDateFormat("yyyy-MM-dd")

case class Register (d: java.util.Date, uuid: String, cust_id: String, lat: Float,
lng: Float)
case class Click (d: java.util.Date, uuid: String, landing_page: Int)

val reg = sc.textFile("reg.tsv").map(_.split("\t")).map(
 r => (r(1), Register(format.parse(r(0)), r(1), r(2), r(3).toFloat, r(4).toFloat))
)

val clk = sc.textFile("clk.tsv").map(_.split("\t")).map(
 c => (c(1), Click(format.parse(c(0)), c(1), c(2).trim.toInt))
)

reg.join(clk).take(2)
```

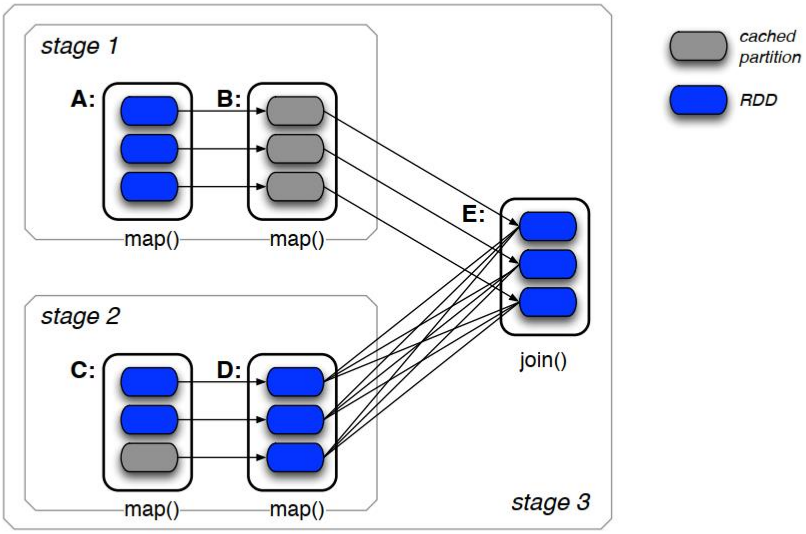


Source Code

- Operator graph 생성

```
scala> reg.join(clk).toDebugString
res5: String =
FlatMappedValuesRDD[46] at join at <console>:23 (1 partitions)
  MappedValuesRDD[45] at join at <console>:23 (1 partitions)
    CoGroupedRDD[44] at join at <console>:23 (1 partitions)
      MappedRDD[36] at map at <console>:16 (1 partitions)
        MappedRDD[35] at map at <console>:16 (1 partitions)
          MappedRDD[34] at textFile at <console>:16 (1 partitions)
            HadoopRDD[33] at textFile at <console>:16 (1 partitions)
      MappedRDD[40] at map at <console>:16 (1 partitions)
        MappedRDD[39] at map at <console>:16 (1 partitions)
          MappedRDD[38] at textFile at <console>:16 (1 partitions)
            HadoopRDD[37] at textFile at <console>:16 (1 partitions)
```

Operator Graph



실습

- Github의 README.md와 CHANGES.txt를 사용
  - 특정 키워드를 가진 라인에 대해서 FILTER하는 RDD 생성
  - 각 라인에 대해 Word Count를 수행
  - 두 RDD간 조인(Join)