

Apache Spark를 활용한 로그분석

빅데이터팀 김정익

INDEX

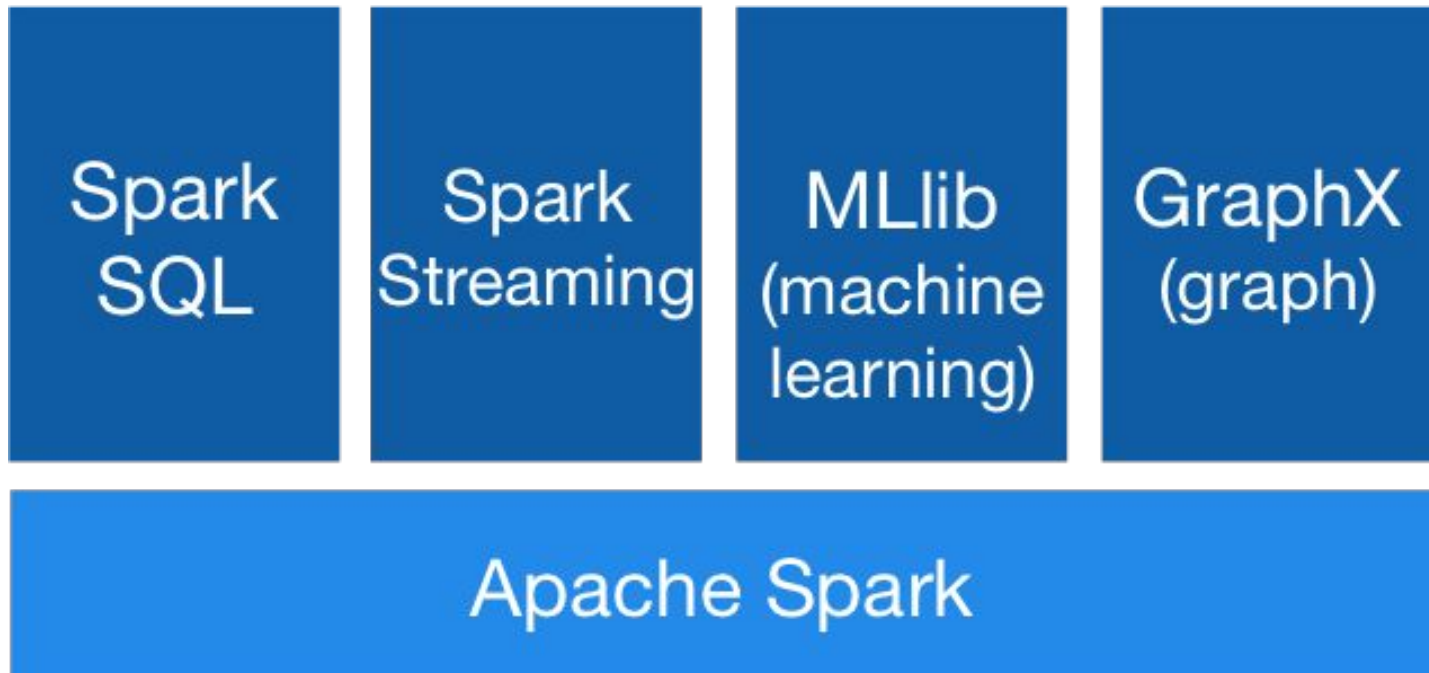
1. 아파치 스파크 개요
2. 분석 환경 및 데이터 소개
3. 데이터 처리 과정
4. 스파크 성능 개선



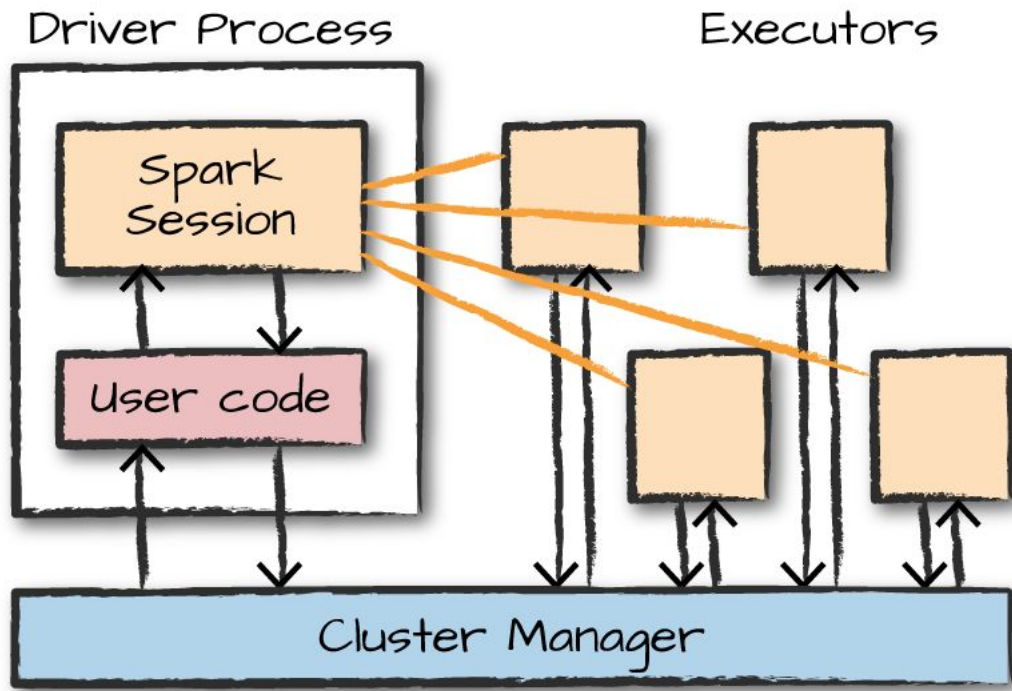
What is SPARK?

- 인메모리 기반 빅데이터 처리 플랫폼
- 맵리듀스보다 100배 빠른속도
- API와 런타임 아키텍처를 이용해 로컬 프로그램을 작성하는 것과 유사한 방식으로 분산 프로그램 작성 가능
- Scala, Python, Java, R 등 다양한 프로그래밍 언어를 지원
- 실시간 데이터 처리 기능, 머신러닝, SQL 연산, 그래프 알고리즘, 일괄 처리 등 다양한 라이브러리를 지원하는 통합플랫폼

Spark Ecosystem



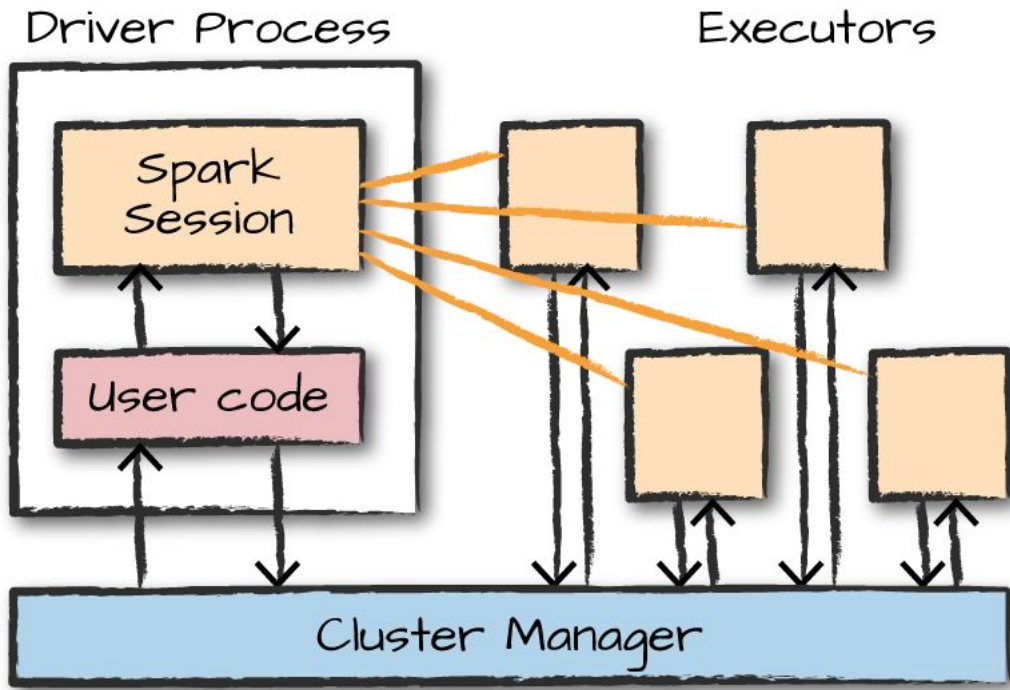
Spark Basic Architecture



Cluster Manager

- Application 자원을 할당, 제거하는 등 Cluster 자원을 관리
- Standalone, Yarn, Mesos

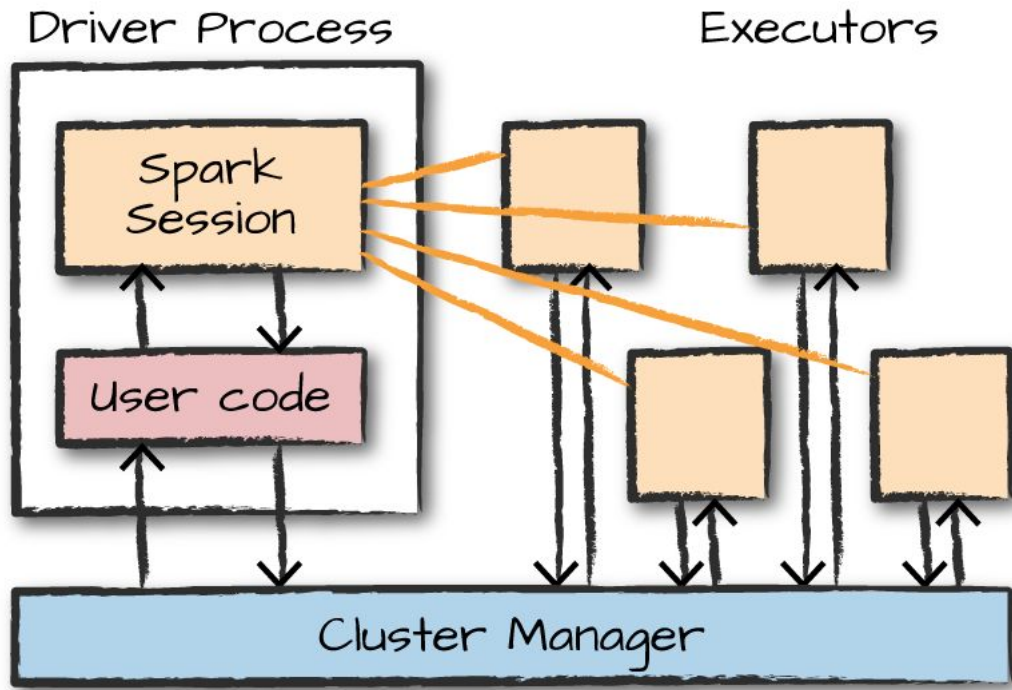
Spark Basic Architecture



Driver Process

- 스파크 어플리케이션을 실행하는 프로세스
- 하나의 어플리케이션마다 하나의 드라이버가 존재
- 클러스터 매니저에 메모리 및 CPU 리소스를 요청
- 애플리케이션 잡을 태스크로 분할하여 Executor로 전달

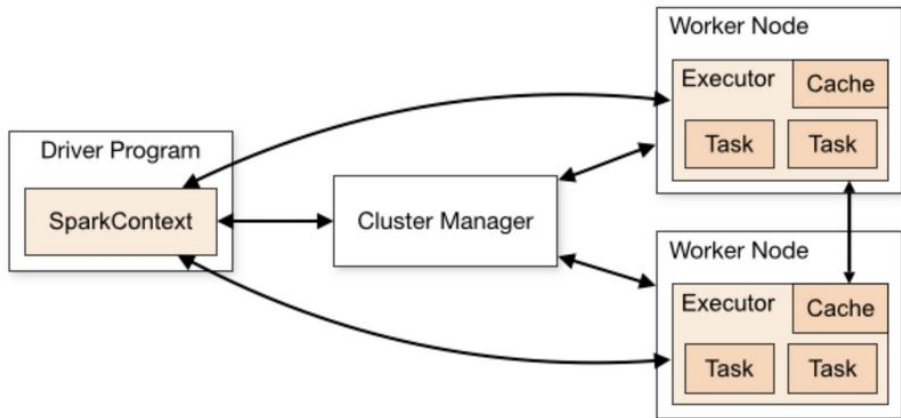
Spark Basic Architecture



Executors

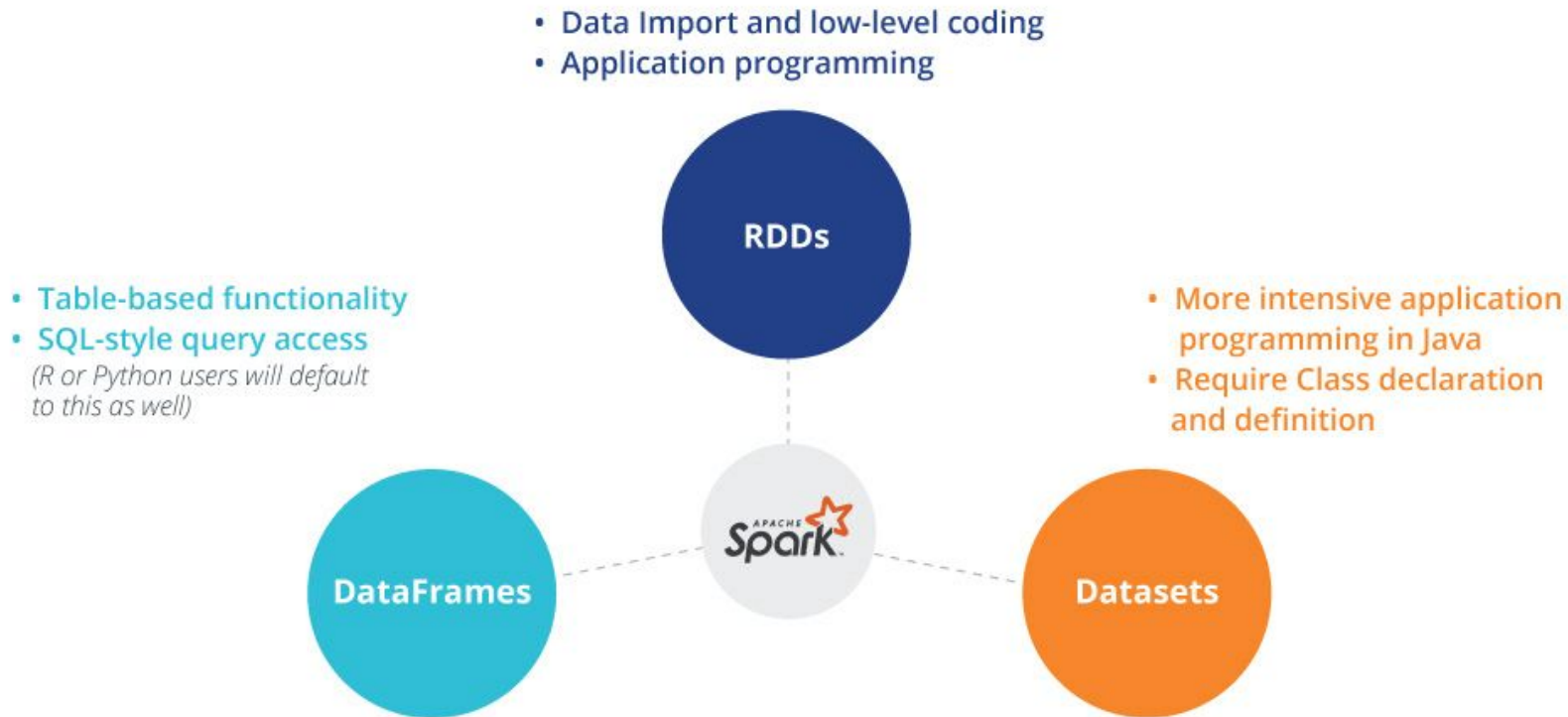
- 드라이버가 요청한 태스크를 받아서 실행
- 태스크를 수행한 결과를 Driver로 전송

스파크 작동순서



1. 사용자가 spark-submit을 사용해 작성한 Application을 실행
2. spark-submit은 Driver Program을 실행하여 main() 메소드 호출
3. Driver에서 생성된 SparkContext는 Cluster Manager로부터 Executor실행을 위한 리소스를 요청
4. Cluster Manager는 Executor를 실행
5. Driver Program은 Application을 Task단위로 나누어 Executor에게 전송
6. Executor는 Task를 실행
7. Executor는 Application이 종료가 되면, 결과를 Driver Program에게 전달하고, Cluster Manager에게 리소스를 반납

Spark API

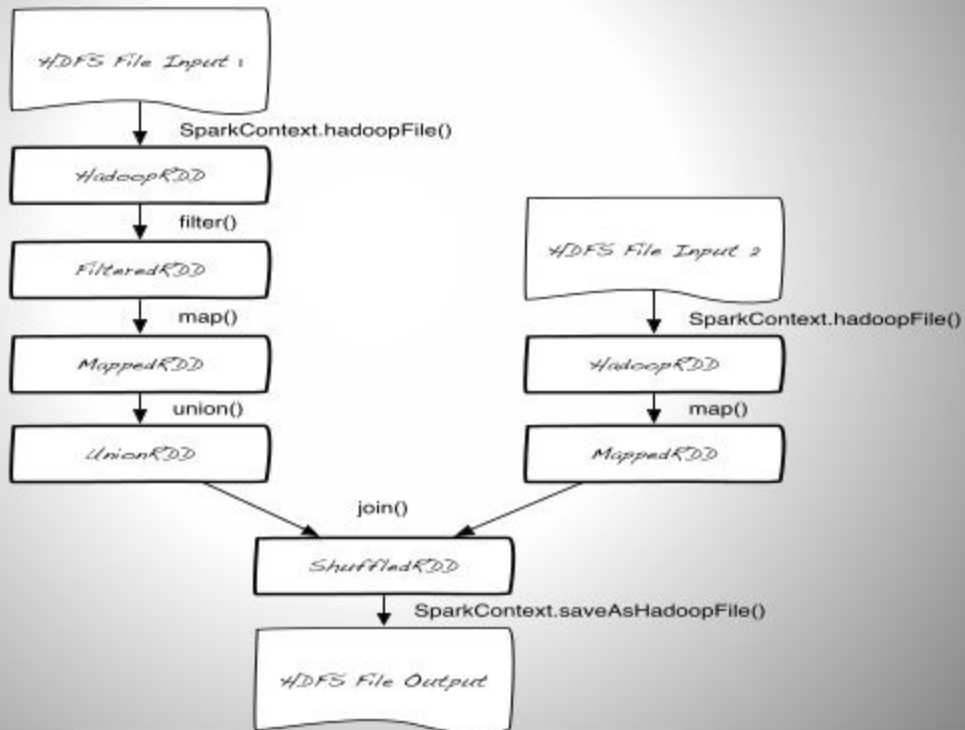


RDD

- Resilient Distributed Data
- Fault tolerance
- RDD Lineage (DAG)

RDD Lineage

RDD Lineage



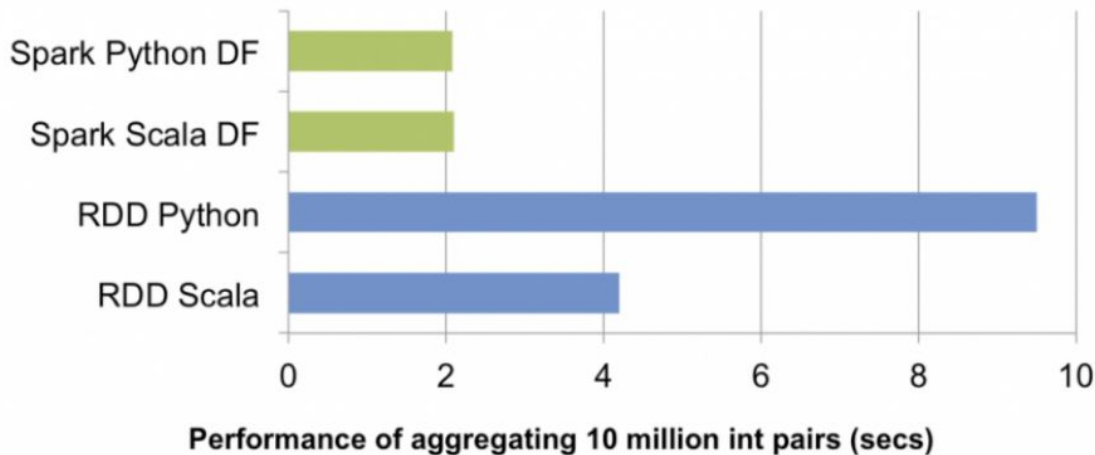
Data

Operation

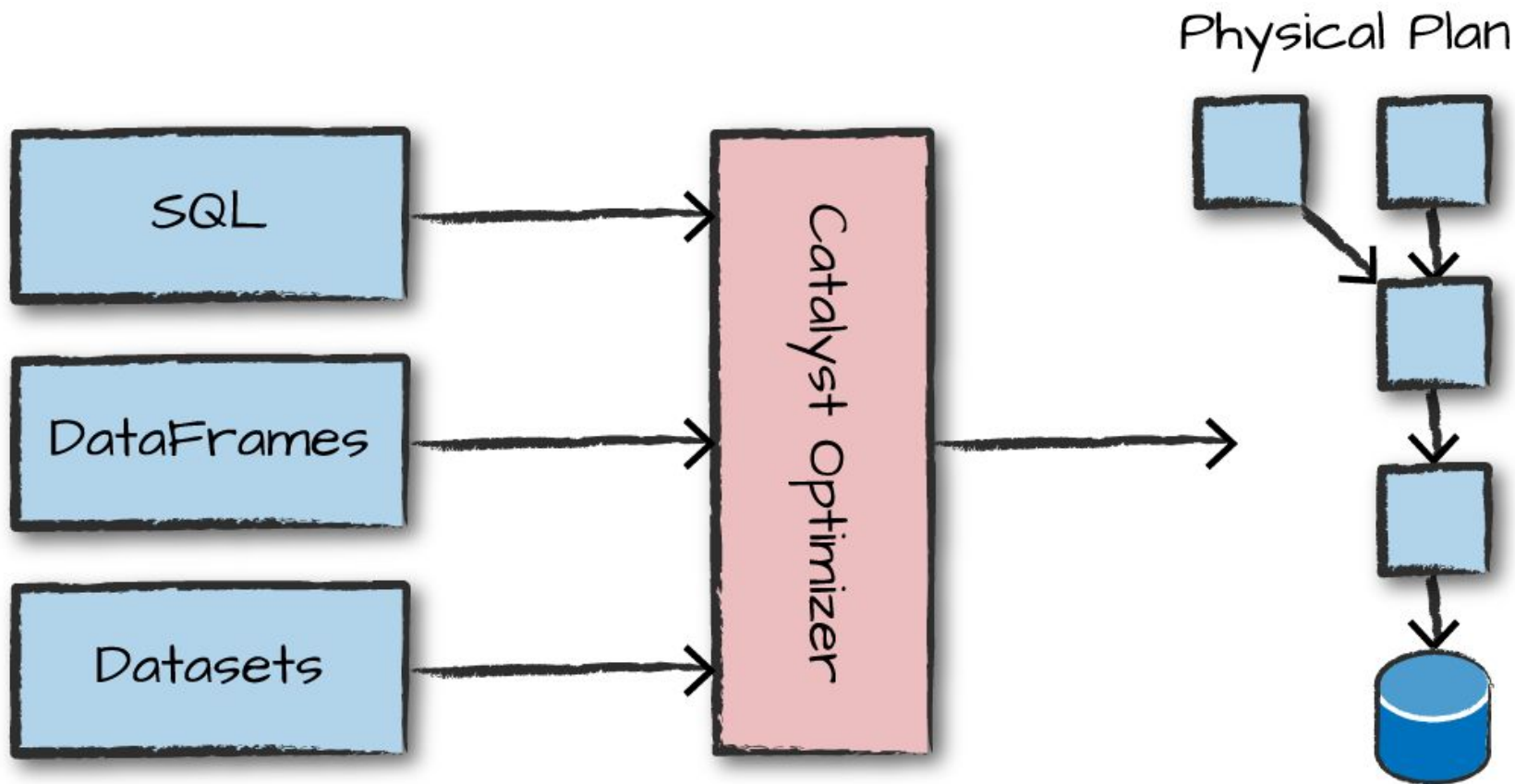
- Transformation
(e.g. map, filter, groupBy, join)
- Action
(e.g. count, collect, save)
- **Lazy Evaluation**

Structured API

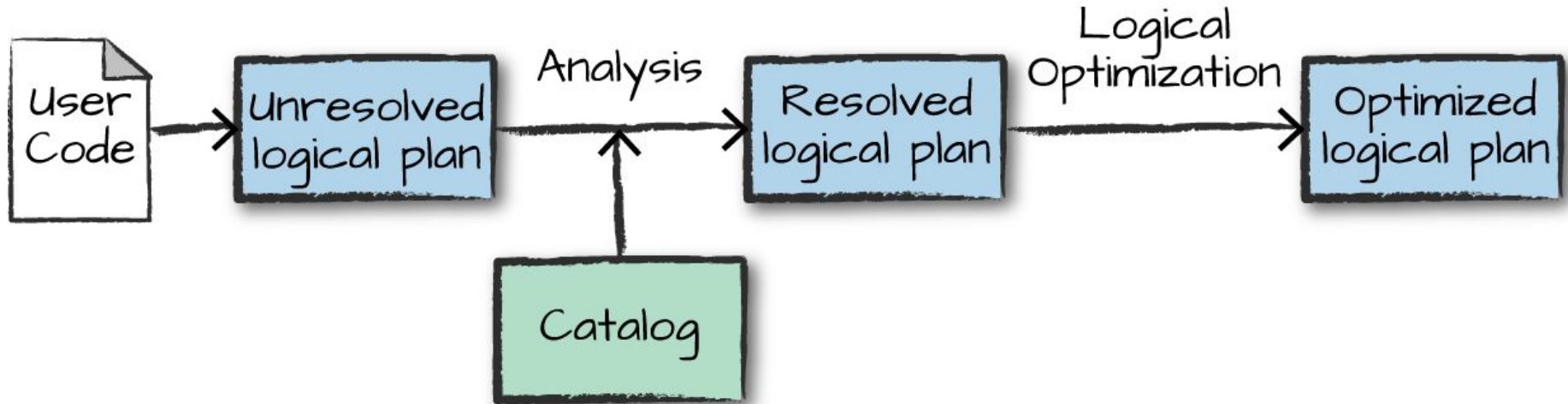
- 고수준의 API
- 스키마, 컬럼으로 구조화 된 데이터
- 옵티마이저에 의한 최적화



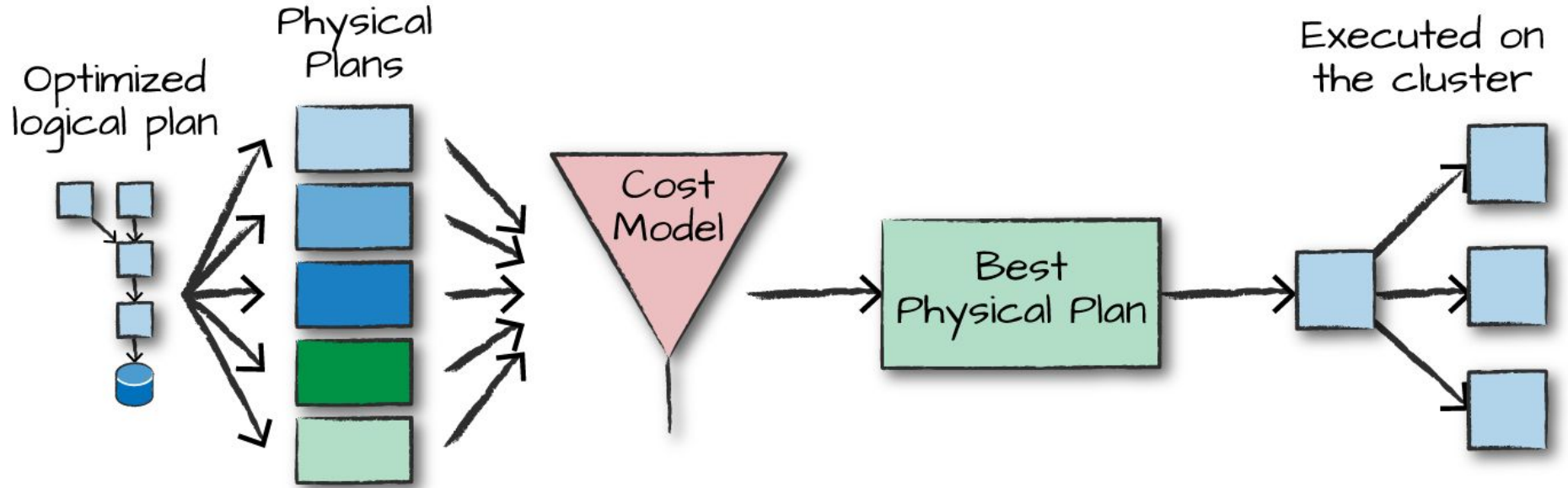
Structured API Execution



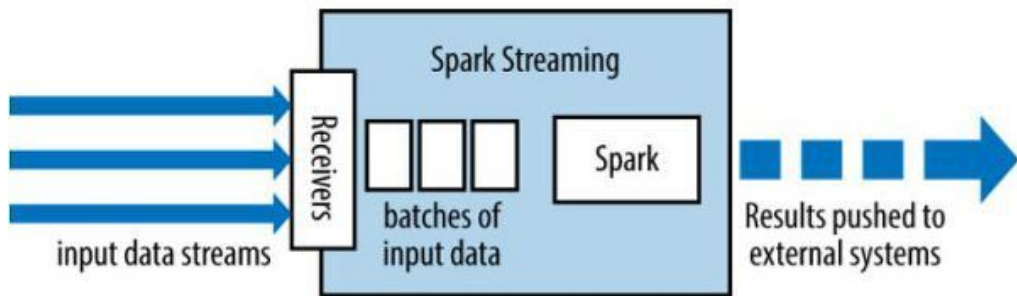
Logical Planning



Physical Planning

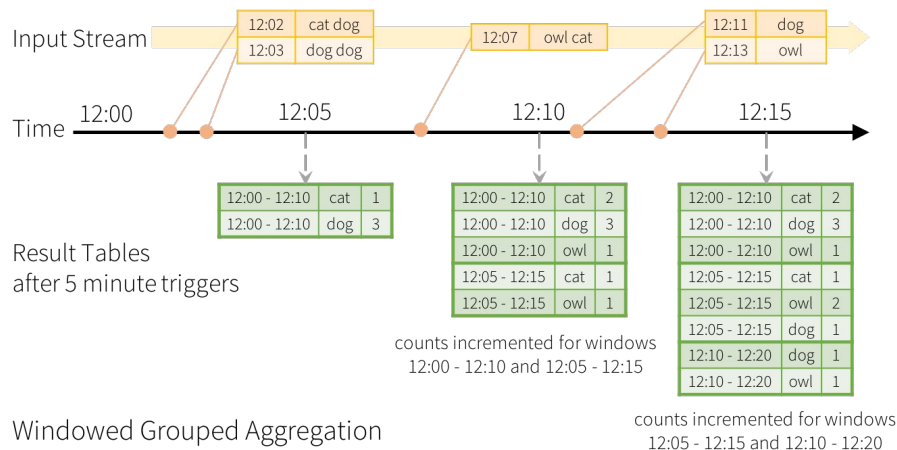


Spark Streaming



- 마이크로 배치 아키텍처로 near real time
- DStream을 생성하여 데이터 처리
- 연속적으로 생성되는 스트림 데이터를 처리하는 API를 제공
- 다양한 데이터 소스, 싱크 제공
- 구조적 스트리밍 API 제공
- 윈도우 타임을 지정한 데이터 처리

Spark Streaming



Windowed Grouped Aggregation
with 10 min windows, sliding every 5 mins

- 마이크로 배치 아키텍처로 near real time
- DStream을 생성하여 데이터 처리
- 연속적으로 생성되는 스트림 데이터를 처리하는 API를 제공
- 다양한 데이터 소스, 싱크 제공
- 구조적 스트리밍 API 제공
- 윈도우 타임을 지정한 데이터 처리

분석과정

host		user_id	date		http_method	url	http_version	status_code	length
referrer	user_agent		cookie						
14.5.4.6	-	-	[01/Jul/2020:14:0...		GET	https://estat.zum...	HTTP/1.1	200	0 https://
zum.com/?...	Mozilla/5.0 (Wind...	TUID=FD30B992-E1...							
14.5.4.6	-	-	[01/Jul/2020:14:1...		GET	https://estat.zum...	HTTP/1.1	200	0 https://
zum.com/?...	Mozilla/5.0 (Wind...	TUID=FD30B992-E1...							
14.5.4.6	-	-	[01/Jul/2020:14:2...		GET	https://estat.zum...	HTTP/1.1	200	0 https://
zum.com/?...	Mozilla/5.0 (Wind...	TUID=FD30B992-E1...							
14.5.4.6	-	-	[01/Jul/2020:14:2...		POST	https://estat.zum...	HTTP/1.1	200	0 https://
zum.com/?...	Mozilla/5.0 (Wind...	TUID=FD30B992-E1...							
14.5.4.6	-	-	[01/Jul/2020:14:3...		GET	https://estat.zum...	HTTP/1.1	200	0 https://
zum.com/?...	Mozilla/5.0 (Wind...	TUID=FD30B992-E1...							
14.6.2.7	-	-	[01/Jul/2020:14:0...		GET	https://estat.zum...	HTTP/1.1	200	0 https://
zum.com/?...	Mozilla/5.0 (Wind...	ZUT=nc%3D5a04817...							
14.6.2.7	-	-	[01/Jul/2020:14:0...		GET	https://estat.zum...	HTTP/1.1	200	0 https://
zum.com/?...	Mozilla/5.0 (Wind...	ZUT=nc%3D5a04817...							
14.6.2.7	-	-	[01/Jul/2020:14:0...		POST	https://estat.zum...	HTTP/1.1	200	0 https://
zum.com/?...	Mozilla/5.0 (Wind...	ZUT=nc%3D5a04817...							
14.6.5.3	-	-	[01/Jul/2020:14:4...		GET	https://estat.zum...	HTTP/1.1	200	0 https://
hub.zum.c...	Mozilla/5.0 (Wind...	TUID=4F02774E-45...							
14.6.5.3	-	-	[01/Jul/2020:14:4...		GET	https://estat.zum...	HTTP/1.1	200	0 https://
hub.zum.c...	Mozilla/5.0 (Wind...	TUID=4F02774E-45...							
14.6.5.7	-	-	[01/Jul/2020:14:3...		POST	https://estat.zum...	HTTP/1.1	200	0 https://
zum.com/?...	Mozilla/5.0 (Wind...								
14.6.8.5	-	-	[01/Jul/2020:14:5...		POST	https://estat.zum...	HTTP/1.1	200	0 https://
zum.com/?...	Mozilla/5.0 (Wind...	TUID=295BC9AF-49...							
49.1.1.5	-	-	[01/Jul/2020:14:1...		POST	https://estat.zum...	HTTP/1.1	200	0 http
s://zum.com/	Mozilla/5.0 (Wind...								

- Cluster 환경 구성

- 2 Cores X 8GB 메모리 X 5 instances

- log-data 15.9 GiB

- 12 Columns

("host", "none", "user_id", "date", "http_method", "url", "http_version", "status_code", "length", "referrer", "user_agent", "cookie")

- 실제 사용할 컬럼

("host", "url", "cookie")

- 결과 - 주식 종목 관련 키워드 검색 수

분석과정

- host가 112.216.127.98 아닌 레코드 삭제

- cookie 에 ZUID 필드값을 포함하는 레코드

```
{
  "event": "@PageView",
  "properties": {
    "ver": 3,
    "puid": "UATAAFAZCNCMRDLo2iHsaAPvPgTE4BZMR-49Iqs",
    "variants": [],
    "mabVariants": [1],
    "mabUpdate": 1591857057577,
    "data": {
      "gnb": "related",
      "error": 0,
      "issue": 0,
      "hd": "21",
      "dd": 7,
      "noresult": 0,
      "l": 0,
      "r": 2,
      "b": 1,
      "f": 1,
      "e": 0,
      "n": 50,
      "m": 10,
      "sv": 0,
      "r": 1,
      "s": 0,
      "spr": 0,
      "a": 0,
      "newscp": 0,
      "feed": {
        "n": "siteLink",
        "r": 1,
        "t": 1
      },
      "n": "contents:개별모의임대물류관리이션",
      "r": 2,
      "n": "site",
      "r": 3,
      "n": "dictionary",
      "r": 4,
      "defaultTab": "wiki",
      "n": "blo",
      "r": 5,
      "selectCate": "all",
      "n": "news",
      "r": 6,
      "selectCate": "all",
      "n": "map",
      "r": 7,
      "defaultTab": "map-place",
      "targete": true,
      "polygon": false,
      "n": "web",
      "r": 8,
      "n": "realtime",
      "r": 9,
      "defaultTab": "realtime:total",
      "n": "board",
      "r": 10,
      "n": "shopping",
      "r": 11,
      "n": "book",
      "r": 12,
      "n": "video",
      "r": 13,
      "n": "image",
      "r": 14,
      "n": "openinternet",
      "r": 15,
      "n": "expert",
      "r": 16,
      "scpData": {
        "cid": 3423504,
        "cateid": 577,
        "referrer": "",
        "url": "http://search.zum.com/search.zum?method=uni&query=%ED%98%88%ED%83%9D%EC%8A%4d&rd=1&qm=f&supp.top&sg_q=%E3%85%E&sg_id=1&sg_tot=30&sg_col=1&sg_row=1&sg_fix=pre&sm=v3lite",
        "screenSize": "1920x1080",
        "clientDocSize": "1167x589",
        "time": 1593581197,
        "version": "A16"
      }
    }
  }
}
```

```

"event": "PageView", "properties": {"ver": "3", "puid": "UA1AAAFzCMLE40Flj_EKQv55XsY1BnrlC1nj8J5p", "variants": {}, "mabVariants": {}, "mabUpdate": "1591857510474", "data": {"gnb": "related", "error": "0", "issue": "0", "hd": "0", "dd": "6", "noresult": "0", "l": "0", "r": "5", "f": "1", "i": "2", "e": "0", "n": "mr", "l": "sv", "0": "tr", "1": "sr", "0": "spr", "0": "a", "0": "newspc": "1", "feed": {"n": "powerlink", "r": "1"}, {"n": "powercontents", "r": "2"}, {"n": "blog", "r": "3"}, {"selectCat": "all"}, {"n": "site", "r": "4"}, {"n": "board", "r": "5"}, {"n": "new", "r": "6"}, {"selectCat": "all"}, {"n": "realtime", "r": "7", "defaultTab": "realtime:total"}, {"n": "image", "r": "8"}, {"n": "web", "r": "9"}, {"n": "openinternet", "r": "10"}, {"n": "video", "r": "11"}, {"n": "book", "r": "12"}, {"n": "expert", "r": "13"}, {"scPvData": {"e": "1", "referrer": "http://search.zum.com/search.zum?method=unioption=accu&r=l&q=f_typing.top&query=%E8%8C80%6B%9C%E6%9A%8C&sm=y", "url": "http://search.zum.com/search.zum?associate_id=1&associate_type=query=&eb8298fec9d498feb8e808ecb698f6c6a1b0&ed88b8c6cmethod=unioption=accu&qgm=g_associate&r=1", "screenSize": "1518x853", "clientDocSize": "1499x666", "time": "1593580945", "version": "A16"}}, 0}

```

[illegible]

```
selected_log = load_log.select("host", "url", "cookie")\
    .where(load_log['host'] != "112.216.127.98")\
    .drop('host')\
    .filter(load_log["cookie"].contains("_ZUID"))
```

base64 decoding 과정

```
status_log = selected_log\
    .withColumn('status', functions.regexp_extract('url', 'data=(.*)', 1))\
    .withColumn('status', functions.regexp_replace('status', '&time(.*)', ''))
```

```
base64_decoded_log = status_log\
    .withColumn('status', unbase64(status_log['status']).cast("string"))\
    .drop("url")
```

- ## 통계 event가 PageView인 레코드

- search zum에 접속한 사람만 필터링

```
event_log = base64_decoded_log\
    .filter(base64_decoded_log['status'].startswith('{ "event": "@PageView" }'))\
    .filter(base64_decoded_log['status'].contains('search.zum.com'))
```

분석과정

```
[{"event": "@PageView", "properties": {"ver": 3, "puid": "UAIAAFzCnCMRDloziHsaApEvPgTE4BZMR-49Iqs", "variants": [], "mabVariants": [], "mabUpdate": 1591857507757, "data": {"gnb": "related", "error": 0, "issue": 0, "hd": 21, "dd": 7, "noresult": 0, "l": 0, "r": 2, "f": 1, "i": 2, "e": 0, "n": 0, "mr": 1, "sv": 0, "tr": 1, "sr": 0, "spr": 0, "a": 0, "news": 0, "feed": [{"n": "sitelink", "r": 1}, {"n": "contents": "개별모바일애플리케이션", "r": 2}, {"n": "site", "r": 3}, {"n": "dictionary", "r": 4, "defaultTab": "wiki"}, {"n": "blog", "r": 5, "selectCate": "all"}, {"n": "news", "r": 6, "selectCate": "all"}, {"n": "map", "r": 7, "defaultTab": "map-place", "targeted": true, "polygon": false}, {"n": "web", "r": 8}, {"n": "realtime", "r": 9, "defaultTab": "realtime:total"}, {"n": "board", "r": 10}, {"n": "shopping", "r": 11}, {"n": "book", "r": 12}, {"n": "video", "r": 13}, {"n": "image", "r": 14}, {"n": "openinternet", "r": 15}, {"n": "expert", "r": 16}], "scPvData": [{"cid": "3423504", "cateid": "577"}], "referrer": "", "url": "http://search.zum.com/search.zum?method=uni&query=%ED%99%88%ED%83%9D%EC%8A%A4&rd=1&qm=f_suggest.top&sug_q=%E3%85%8E&sug_id=1&sug_tot=30&sug_col=1&sug_row=1&sug_fix=pre&sm=v3lite", "screenSize": "1920x1080", "clientDocSize": "1167x589", "time": 1593581197, "version": "A16"}}]

[{"event": "@PageView", "properties": {"ver": 3, "puid": "UAIAAFzCMLE40fLj_EKQvS5XsYIBnlrC1Nj8J5p", "variants": [], "mabVariants": [], "mabUpdate": 1591857510474, "data": {"gnb": "related", "error": 0, "issue": 0, "hd": 0, "dd": 6, "noresult": 0, "l": 0, "r": 2, "f": 1, "i": 2, "e": 0, "n": 0, "mr": 1, "sv": 0, "tr": 1, "sr": 0, "spr": 0, "a": 0, "news": 0, "feed": [{"n": "powerlink", "r": 1}, {"n": "contents": "powercontents", "r": 2}, {"n": "blog", "r": 3, "selectCate": "all"}, {"n": "site", "r": 4}, {"n": "board", "r": 5}, {"n": "new", "r": 6, "selectCate": "all"}, {"n": "realtime", "r": 7, "defaultTab": "realtime:total"}, {"n": "image", "r": 8}, {"n": "web", "r": 9}, {"n": "openinternet", "r": 10}, {"n": "video", "r": 11}, {"n": "book", "r": 12}, {"n": "expert", "r": 13}], "scPvData": [{"referrer": "http://search.zum.com/search.zum?method=uni&option=accu&rd=1&qm=f_typing.top&query=%EB%8C%80%EC%B6%9C%EC%A1%B0%EB%80%EC%B6%9C%EC%A1%B0%ED%83%9D%EC%8A%A4&rd=1", "screenSize": "1518x853", "clientDocSize": "1499x666", "time": 1593580945, "version": "A16"}}]

[{"event": "@PageView", "properties": {"ver": 3, "puid": "UAIAAFzCMWbxEcsTka_Et1KRgAZrClQh5pgce", "variants": [], "mabVariants": [], "mabUpdate": 1591857510474, "data": {"gnb": "blank", "error": 0, "issue": 0, "hd": 0, "dd": 0, "l": 0, "r": 0, "f": 1, "i": 2, "e": 0, "n": 0, "mr": 1, "sv": 0, "tr": 1, "sr": 0, "spr": 0, "a": 0, "news": 0, "feed": [{"n": "blog", "r": 1, "selectCate": "all"}, {"n": "news", "r": 2, "selectCate": "all"}, {"n": "board", "r": 3}, {"n": "site", "r": 4}, {"n": "image", "r": 5}, {"n": "web", "r": 6}, {"n": "openinternet", "r": 7}, {"n": "expert", "r": 8}], "scPvData": [{"referrer": "http://search.zum.com/search.zum?method=uni&query=%EB%82%98%EC%9D%98%EB%8C%80%EC%B6%9C%EB%82%B4%EC%97%AD%EC%A1%B0%ED%9A%8C&rd=1&qm=f_suggest.top&sug_q=%EB%82%98%EC%9D%98%EB%8C%80%EC%B6%9C&sug_id=5&sug_tot=9&sug_col=1&sug_row=5&sug_fix=pre", "screenSize": "1518x853", "clientDocSize": "1499x666", "time": 1593581131, "version": "A16"}}]
```

● 접속한 url에서 query를 추출

```
event_query_log = event_log\
    .withColumn('status', functions.regexp_replace('status', '(.+)"url":', ''))\
    .withColumn('status', functions.regexp_replace('status', '(.+)"query=', ''))\
    .withColumn('status', functions.regexp_replace('status', ' [&"].*', ''))
```

● UDF 함수 정의

```
def decode(val):
    return urllib.parse.unquote(val.encode('utf-8'))

decode_udf = udf(decode, StringType())
```

● 이후 과정

```
cleaned_zum_log = event_query_log.withColumn("status", decode_udf("status"))\
    .withColumn("status", functions.regexp_replace("status", " ", ""))\
    .withColumn("status", functions.regexp_replace("status", "+", ""))\
    .filter(event_query_log.status != "")\
    .withColumnRenamed("status", "query")\
    .withColumn("cookie", split("cookie", "_ZUID=").getItem(1))\
    .withColumn("cookie", split("cookie", ";").getItem(0))\
    .withColumnRenamed("cookie", "id")\
    .dropDuplicates(['id', 'query'])\
    .sort("query")
```


cleaned_zum_log.show()

id	query
FFFFF77E-79EB-4A2...	세라비트
FFFFEB1A-69C9-457...	피자나라치킨공주
FFFFEB1A-69C9-457...	잠옷세트
FFFFEB1A-69C9-457...	영양사도우미
FFFFEB1A-69C9-457...	김현미장관거짓말
FFFFE3DE-7743-498...	낙산비치
FFFF4510-08DE-432...	이순재
FFFFE5546-9A61-4FD...	아이즈원
FFFDE2BA-A88E-4FC...	마켓컬리
FFFDA0DF-FEE5-4B3...	KB국민은행
FFFC87DD-1E1D-419...	clildu.com
FFFC05C9-67E6-4D6...	국민은행개인인터넷뱅킹
FFFBF0D9-99AC-499...	4대보험통합징수포털
FFFB3DDE-9E58-438...	네이트
FFF94117-076B-45D...	아이쿠안전교육
FFF94117-076B-45D...	유튜브
FFF85722-13C0-420...	나이스
FFF75930-1933-477...	핸드폰문자pc로옮기기
FFF65689-8AA8-45D...	부동산증여세계산법
FFF5E38B-2C54-49F...	네이버

only showing top 20 rows

- 접속한 url에서 query를 추출

```
event_query_log = event_log\
    .withColumn('status', functions.regexp_replace('status', '(.)url:', ''))\
    .withColumn('status', functions.regexp_replace('status', '(.)query=', ''))\
    .withColumn('status', functions.regexp_replace('status', ' [&].*', ''))
```

- UDF 함수 정의

```
def decode(val):
    return urllib.parse.unquote(val.encode('utf-8'))
decode_udf = udf(decode, StringType())
```

- 이 후 과정

```
cleaned_zum_log = event_query_log.withColumn("status", decode_udf("status"))\
    .withColumn("status", functions.regexp_replace("status", " ", ""))\
    .withColumn("status", functions.regexp_replace("status", "\\\+", ""))\
    .filter(event_query_log.status != "")\
    .withColumnRenamed("status", "query")\
    .withColumn("cookie", split("cookie", "_ZUID=").getItem(1))\
    .withColumn("cookie", split("cookie", ";").getItem(0))\
    .withColumnRenamed("cookie", "id")\
    .dropDuplicates(['id', 'query'])\
    .sort("query")
```

company_data

company	synonym
동화약품	동화약품
동화약품	000020
KR모터스	에쓰엔티모터스
KR모터스	효성기계공업
KR모터스	000040
KR모터스	에쓰엔티모터스
KR모터스	S&T모터스
KR모터스	에스엔티모터스
KR모터스	KR모터스
KR모터스	에스엔티모터스
경방	경방
경방	000050
메리츠화재	메리츠화재
메리츠화재	동양화재해상보험
메리츠화재	000060
메리츠화재	동양화재
메리츠화재	메리츠보험
메리츠화재	메리츠화재보험
메리츠화재	동양화재보험
삼양홀딩스	000070

suffix

suffix
공시
매출
실적
이익
종목
주가
주식

- Import Company DataFrame
 - company("market", "company", "code", "synonym")
 - suffix("suffix")

- Company DF의 **synonym col + suffix** 가 zum_log의 query와 일치하는 데이터만 추출

- Join

```
joined_company = selected_company.select("company", "synonym")\
    .crossJoin(suffix)\
    .select("company", functions.concat("synonym", "suffix").alias("join"))\
    .union(selected_company)\
    .sort("company", "join")
```

- broadcast join

```
zum_log_joined_company = cleaned_zum_log\
    .join(broadcast(joined_company)\
        , cleaned_zum_log['query'] == joined_company['join'])\
    .drop("query", "join")
```

- group by

```
zum_log_output = zum_log_joined_company\
    .groupBy("company").count().orderBy("count", ascending=False)
```

- save data

```
zum_log_output.coalesce(1).write.format("csv")\
    .option("header", "false")\
    .mode("append")\
    .save(output_path)
```

join_company

company	join
3S	060310
3S	060310공시
3S	060310매출
3S	060310실적
3S	060310이익
3S	060310종목
3S	060310주가
3S	060310주식
3S	3S
3S	3S공시
3S	3S매출
3S	3S실적
3S	3S이익
3S	3S종목
3S	3S주가
3S	3S주식
3S	쓰리에스
3S	쓰리에스공시
3S	쓰리에스매출
3S	쓰리에스실적

- Import Company DataFrame
 - company("market", "company", "code", "synonym")
 - suffix("suffix")

- Company DF의 **synonym col + suffix** 가 zum_log의 query와 일치하는 데이터만 추출

- Join

```
joined_company = selected_company.select("company", "synonym")\
    .crossJoin(suffix)\
    .select("company", functions.concat("synonym", "suffix").alias("join"))\
    .union(selected_company)\
    .sort("company", "join")
```

- broadcast join

```
zum_log_joined_company = cleaned_zum_log\
    .join(broadcast(joined_company)\
        , cleaned_zum_log['query'] == joined_company['join'])\
    .drop("query", "join")
```

- group by

```
zum_log_output = zum_log_joined_company\
    .groupBy("company").count().orderBy("count", ascending=False)
```

- save data

```
zum_log_output.coalesce(1).write.format("csv")\
    .option("header", "false")\
    .mode("append")\
    .save(output_path)
```


분석 결과

```
NAVER,3806
카카오,2421
신한지주,335
KB금융,305
하나금융지주,281
크리스탈,193
기업은행,182
삼성카드,71
아프리카TV,71
다나와,61
삼성전자,56
DGB금융지주,55
인터파크,49
인터파크홀딩스,49
현대차,42
파미셀,38
지코,35
삼성생명,34
키움증권,31
셀트리온,28
메리츠화재,28
넷마블,27
```

- Import Company DataFrame
 - company("market", "company", "code", "synonym")
 - suffix("suffix")

- Company DF의 **synonym col + suffix** 가 zum_log의 query와 일치하는 데이터만 추출

- Join

```
joined_company = selected_company.select("company", "synonym")\
    .crossJoin(suffix)\
    .select("company", functions.concat("synonym", "suffix").alias("join"))\
    .union(selected_company)\
    .sort("company", "join")
```

- broadcast join

```
zum_log_joined_company = cleaned_zum_log\
    .join(broadcast(joined_company)\
        , cleaned_zum_log['query'] == joined_company['join'])\
    .drop("query", "join")
```

- group by

```
zum_log_output = zum_log_joined_company\
    .groupBy("company").count().orderBy("count", ascending=False)
```

- save data

```
zum_log_output.coalesce(1).write.format("csv")\
    .option("header", "false")\
    .mode("append")\
    .save(output_path)
```

Performance Tuning

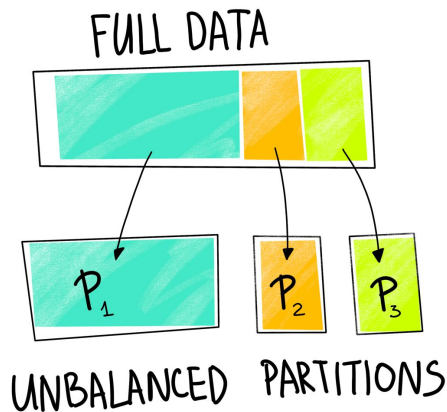
- partitioning
- skewed data issue
- query optimization

- 1 core = 1 task = 1 partition
- shuffle partition
 - ▲ : Task 수가 늘어나 실행 시간이 증가될 수 있고, 드라이버에 오버헤드가 발생할 수 있으나 Shuffle Spill을 예방할 수 있다.
 - ▼ : Task 수가 줄어들면 병렬성이 떨어질 수 있고, Task당 필요 메모리가 높아져 Memory Limit Over (Shuffle spill) 발생 시 성능 저하가 일어날 수 있다.
- 일반적으로 Shuffle Partition의 크기가 100~200MB 가 나을 수 있도록 spark.sql.shuffle.partitions 수를 조절

- **partitioning**

- skewed data issue
- query optimization

- 필터링 과정에서 데이터가 한쪽으로 쏠리는 현상
- Out Of Memory 발생 가능성
- repartition, coalesce 등 파티셔닝 방법을 사용하여 고른 데이터 분포를 가지도록 한다.



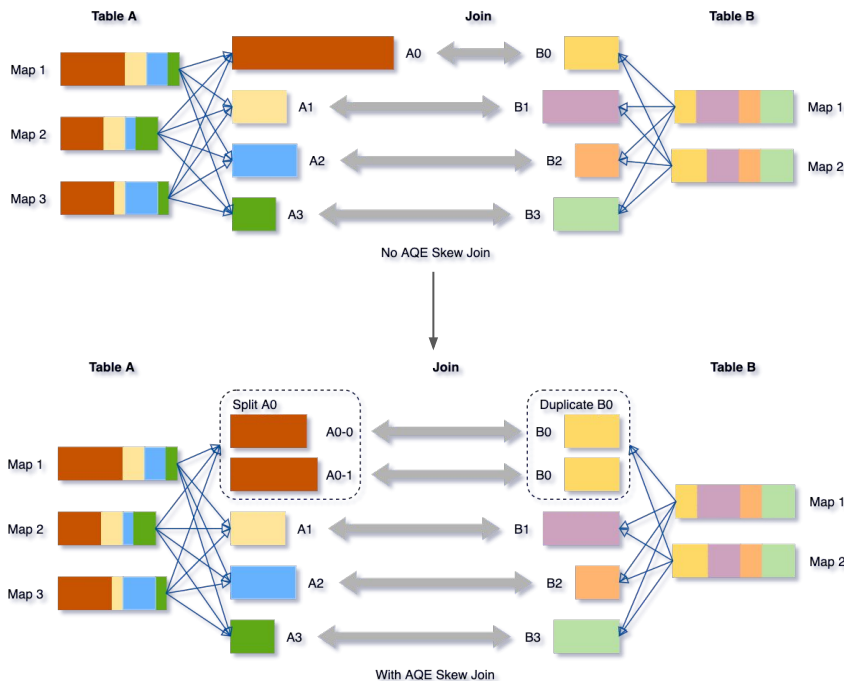
@luminousmen.com

- partitioning
- **skewed data issue**
- query optimization

AQE (Adaptive Query Execution)

- runtime에 더 나은 실행계획을 생성(reoptimizing and adjusting)해 성능을 향상하고 튜닝을 단순화

Dynamically optimizing skew joins



- partitioning

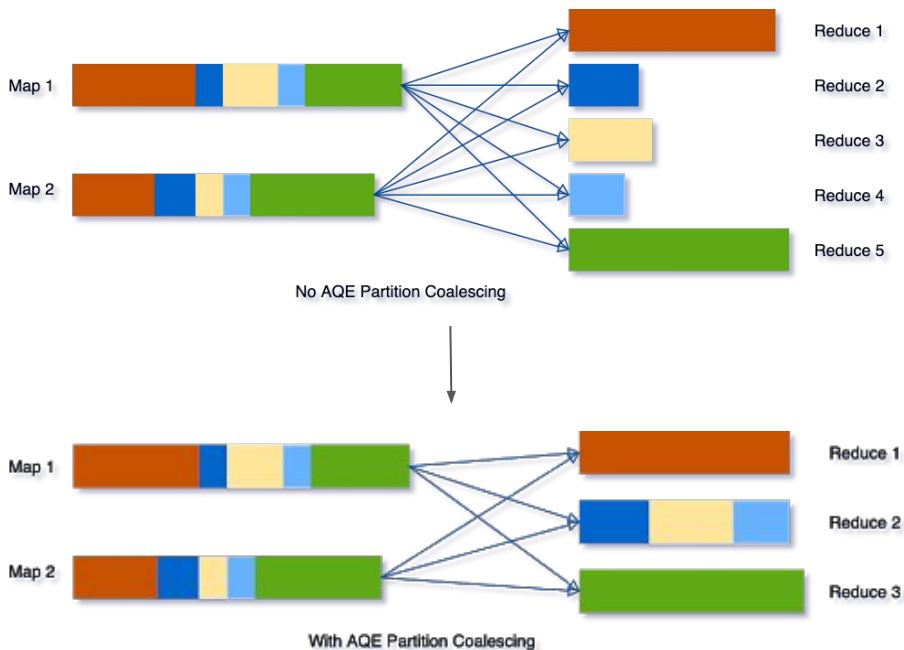
- skewed data issue**

- query optimization

AQE (Adaptive Query Execution)

- runtime에 더 나은 실행계획을 생성(reoptimizing and adjusting)해 성능을 향상하고 튜닝을 단순화

Dynamically coalescing shuffle partitions



- partitioning

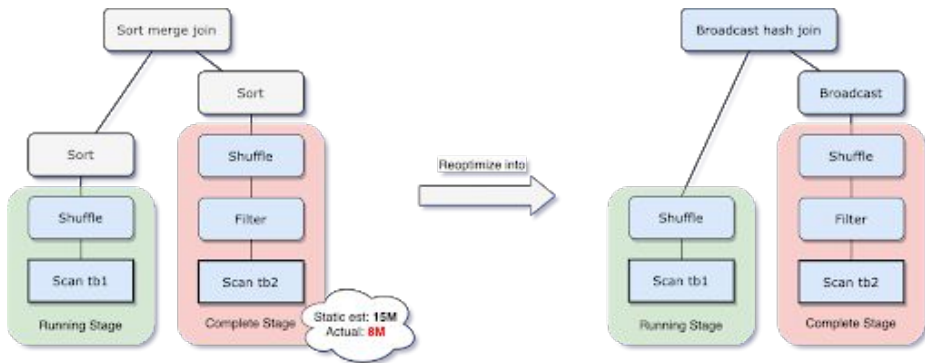
- skewed data issue**

- query optimization

AQE (Adaptive Query Execution)

- runtime에 더 나은 실행계획을 생성(reoptimizing and adjusting)해 성능을 향상하고 튜닝을 단순화

Dynamically switching join strategies



- partitioning

- **skewed data issue**

- query optimization

- **shuffling** 과정 전 query 수행
filtering, aggregation로 shuffle I/O 감소
- **broadcast join**
 - 작은 데이터셋을 join을 수행할 모든 executor에 복사
 - Shuffle과정이 없어 성능 증가
- **caching**
 - 반복적으로 Action이 수행되는 데이터셋을 재사용하기 위해 메모리에 저장

Executor의 memory 구조

- Storage 메모리: Spark의 Cache 데이터 저장을 위해 사용
- Execution 메모리: Shuffle, Join, Sort, Aggregation 등의 연산 과정에서 임시 데이터 저장을 위해 사용

- partitioning

- skewed data issue

- **query optimization**

- shuffle
filterin

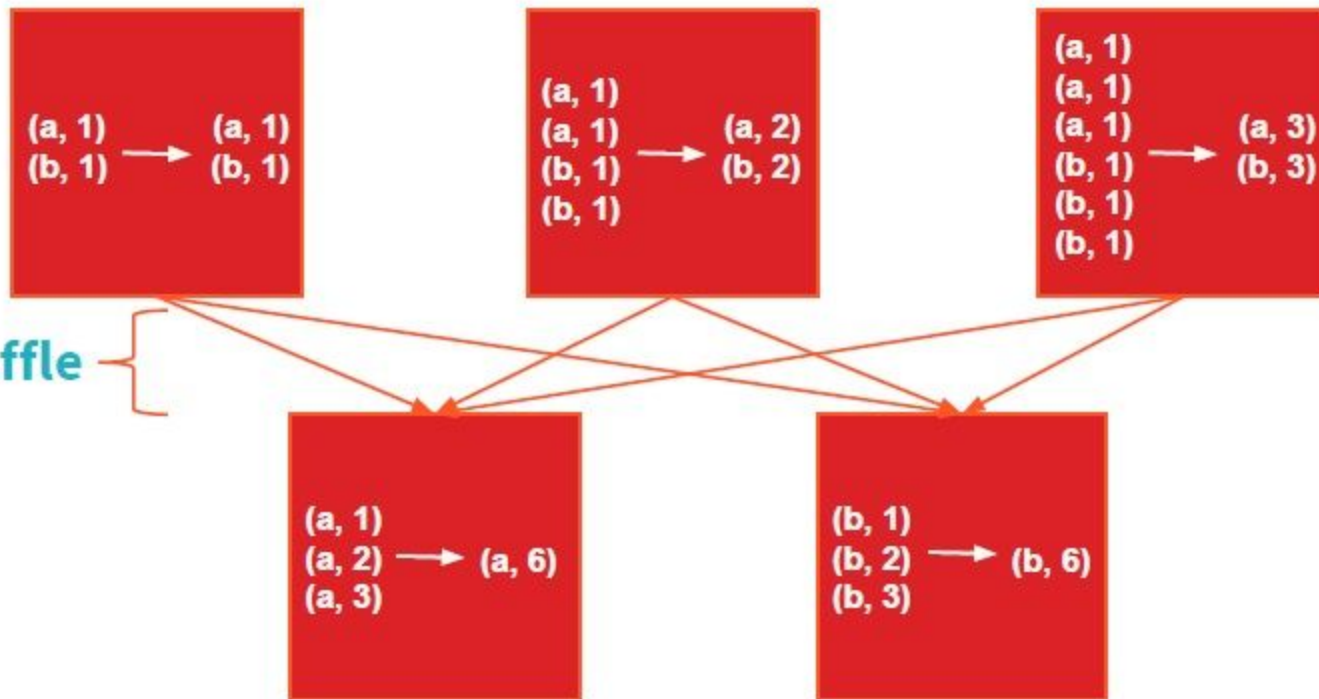
ReduceByKey: Shuffle Step

- broad
- 작은
복사
- Shuffle

- cachi
- 반복
재사용

Execu
- Stora
위해 시
- Execu
Aggreg
저장을

Shuffle



- **shuffling** 과정 전 **query** 수행
filtering, aggregation로 shuffle I/O 감소
- **broadcast join**
 - 작은 데이터셋을 join을 수행할 모든 executor에 복사
 - Shuffle과정이 없어 성능 증가
- **caching**
 - 반복적으로 Action이 수행되는 데이터셋을 재사용하기 위해 메모리에 저장

Executor의 memory 구조

- Storage 메모리: Spark의 Cache 데이터 저장을 위해 사용
- Execution 메모리: Shuffle, Join, Sort, Aggregation 등의 연산 과정에서 임시 데이터 저장을 위해 사용

- partitioning

- skewed data issue

- **query optimization**

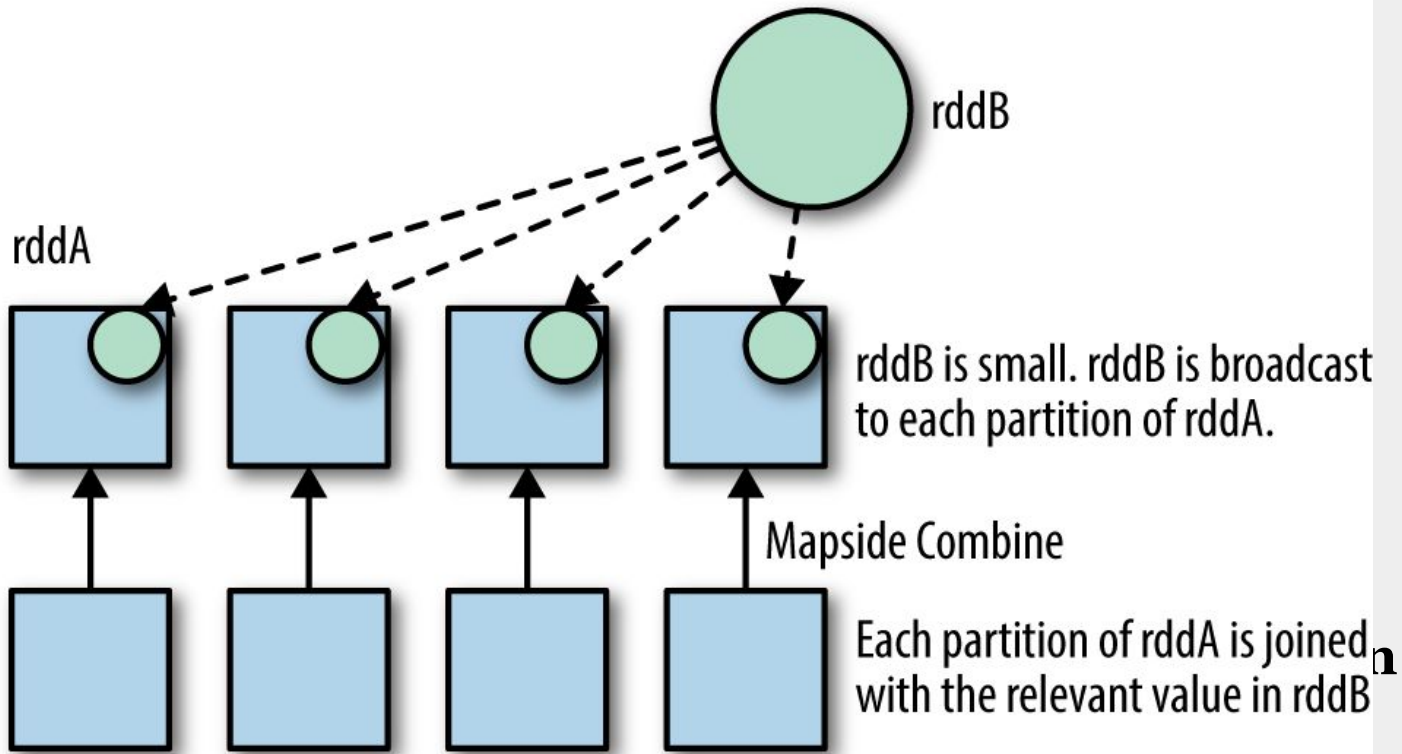
- shuffle
filtering

- broadcast
- 작은 데이터 복사
- Shuffle

- cache
- 반복적 재사용

Execution

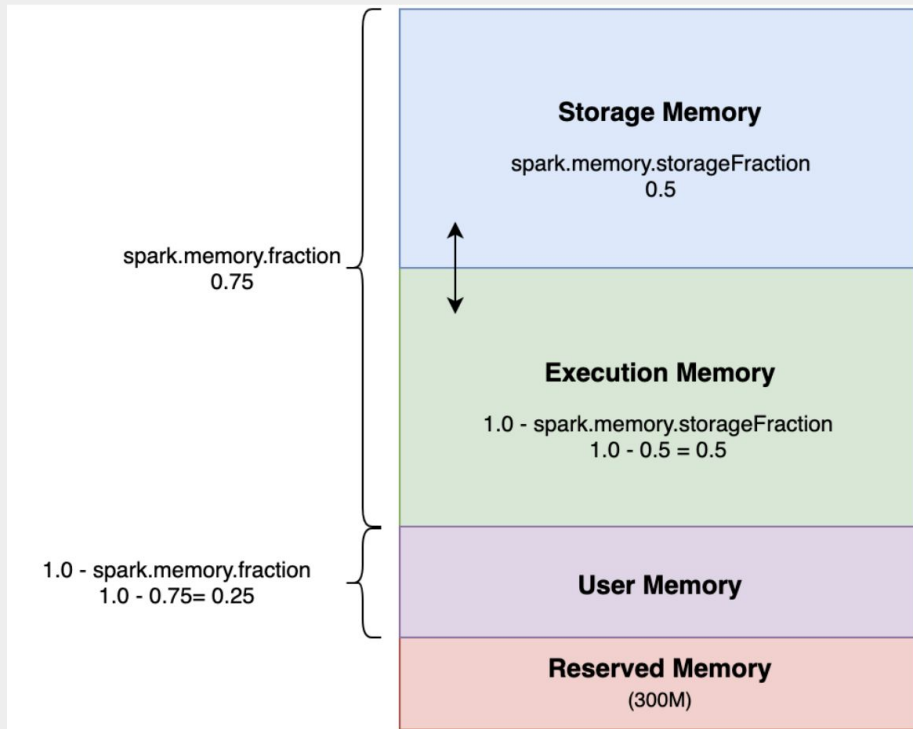
- Storage 위해 사
- Execution Aggrega
- 저장할 수



- **shuffling 과정 전 query 수행**
filtering, aggregation로 shuffle I/O 감소
- **broadcast join**
 - 작은 데이터셋을 join을 수행할 모든 executor에 복사
 - Shuffle과정이 없어 성능 증가
- **caching**
 - 반복적으로 Action이 수행되는 데이터셋을 재사용하기 위해 메모리에 저장

Executor의 memory 구조

- Storage 메모리: Spark의 Cache 데이터 저장을 위해 사용
- Execution 메모리: Shuffle, Join, Sort, Aggregation 등의 연산 과정에서 임시 데이터 저장을 위해 사용



감사합니다.

Q & A