



AWS Glue ETL パフォーマンス・チューニング① 基礎知識編

AWS Black Belt Online Seminar

林田 千瑛
Solution Architect
2021/08



AWS Black Belt Online Seminarとは

- 「サービス別」「ソリューション別」「業種別」のそれぞれのテーマに分け、アマゾン ウェブ サービス ジャパン株式会社が主催するオンラインセミナーシリーズです。
- AWSの技術担当者が、AWSの各サービスについてテーマごとに動画を公開します
- お好きな時間、お好きな場所でご受講いただけるオンデマンド形式です
- 動画を一時停止・スキップすることで、興味がある分野・項目だけの聴講も可能、スキマ時間の学習にもお役立ていただけます

内容についての注意点

- 本資料では2021年8月時点のサービス内容および価格についてご説明しています。最新の情報はAWS公式ウェブサイト(<http://aws.amazon.com>)にてご確認ください。
- 資料作成には十分注意しておりますが、資料内の価格とAWS公式ウェブサイト記載の価格に相違があった場合、AWS公式ウェブサイトの価格を優先とさせていただきます。
- 価格は税抜表記となっております。
日本居住者のお客様には別途消費税をご請求させていただきます。
- AWS does not offer binding price quotes. AWS pricing is publicly available and is subject to change in accordance with the AWS Customer Agreement available at <http://aws.amazon.com/agreement/>. Any pricing information included in this document is provided only as an estimate of usage charges for AWS services based on certain information that you have provided. Monthly charges will be based on your actual use of AWS services, and may vary from the estimates provided.

自己紹介

林田 千瑛 (Chie Hayashida)

アマゾンウェブサービスジャパン
ソリューションアーキテクト



はじめに

- 本セッションは、
『**AWS Glue ETL パフォーマンス・チューニング**』
シリーズの前半、『**基礎知識編**』です。
- 後半の、『**チューニングパターン編**』に続きます。
- 前編となる本セミナーでは、AWS Glue ETLのパフォーマンスチューニングのコツを理解するための基礎知識を解説します

本セミナーの対象者

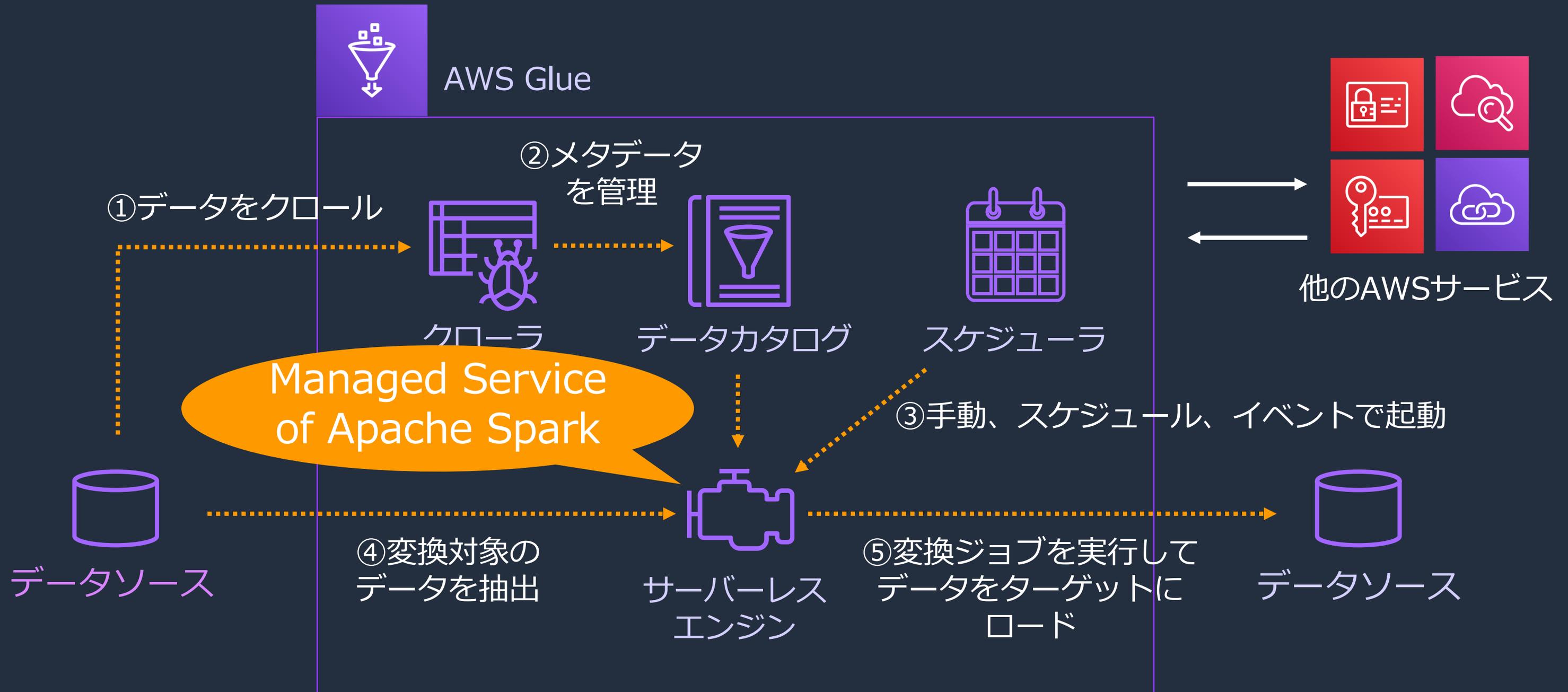
- AWS GlueのAWS Black Belt Online Seminarの内容が理解できている
- Apache Sparkのアプリケーションを書いたことがある
- 今あるAWS Glueジョブを改善したい

※ AWS Glueのお客様のうち多くがPySparkをご利用のため、本資料ではコードはPySparkで書いています

アジェンダ

- Apache Sparkのアーキテクチャ
- AWS Glue固有の機能（パフォーマンスに関連するもの）
- AWS Glue ETL のパフォーマンスチューニングの進め方

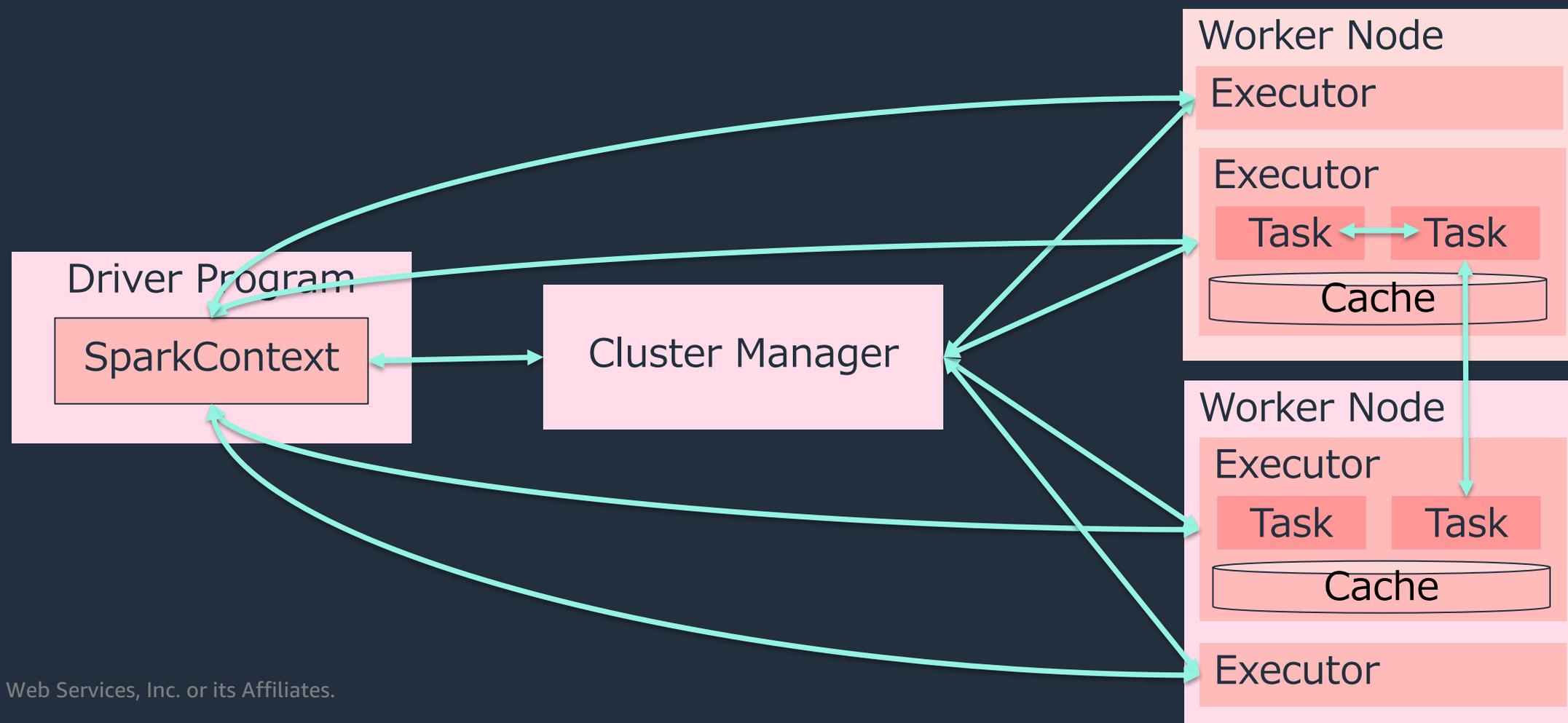
AWS Glue と Apache Spark



Apache Sparkのアーキテクチャ

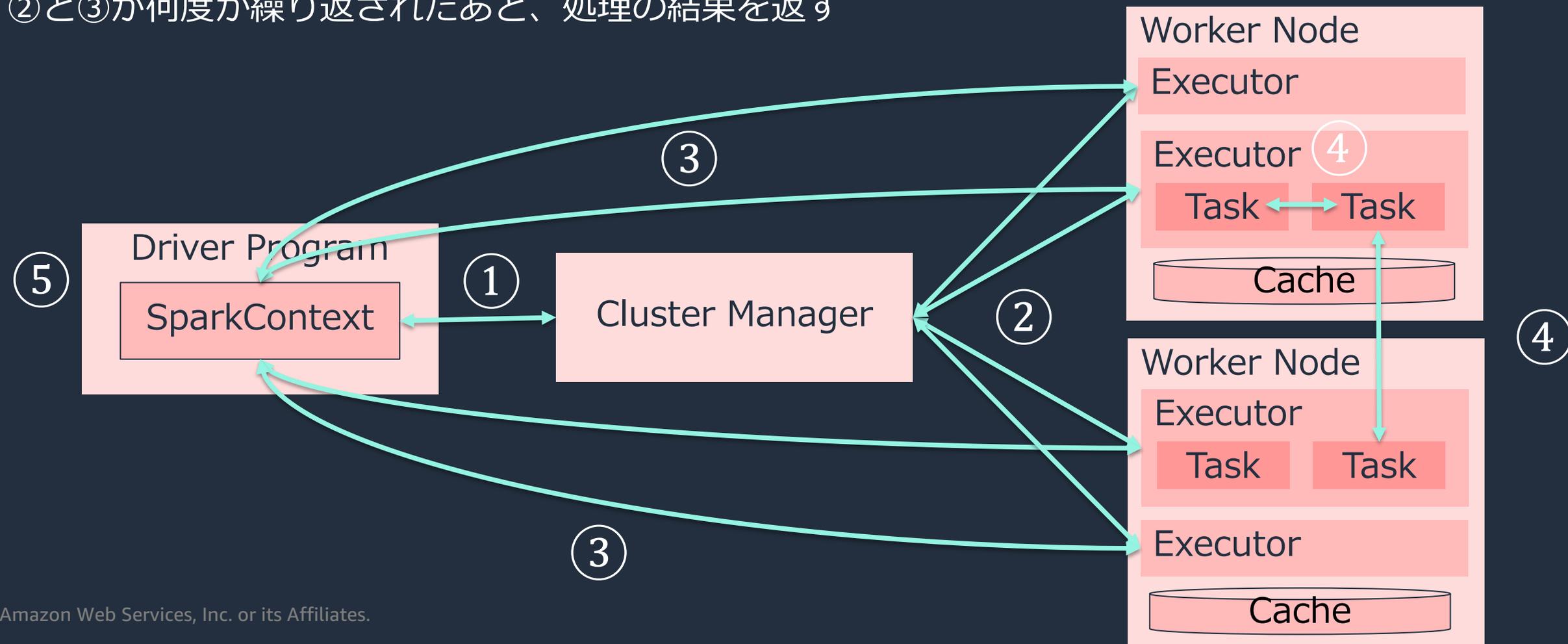
Apache Spark (クラスタモード) のアーキテクチャ

- Cluster Managerは、ジョブを1つ以上のタスクに分割してExecutorに割り当てる
- 1つのワーカノード上では、1つ以上のExecutorが起動する
- 1つのExecutor上では、1つ以上のタスクが実行される



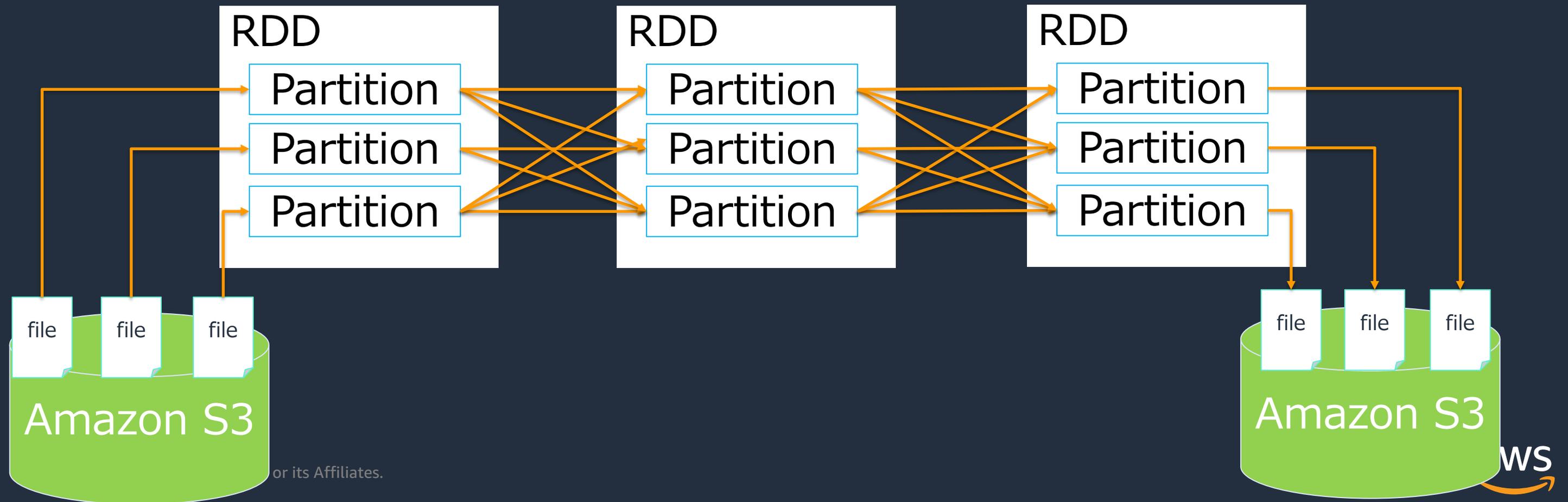
Spark (クラスタモード) のアーキテクチャ

- ① : アプリケーションで必要となるリソースを要求する
- ② : 各ワーカー上でジョブに必要なExecutorを起動する
- ③ : 処理をタスクに分割してExecutorに割り当てる
- ④ : 各Executorに対してタスクを割り当てる。タスクが終わったらExecutorはDriver Programにそれを伝える
- ⑤ : 必要に応じてタスク同士でデータのやりとりをする

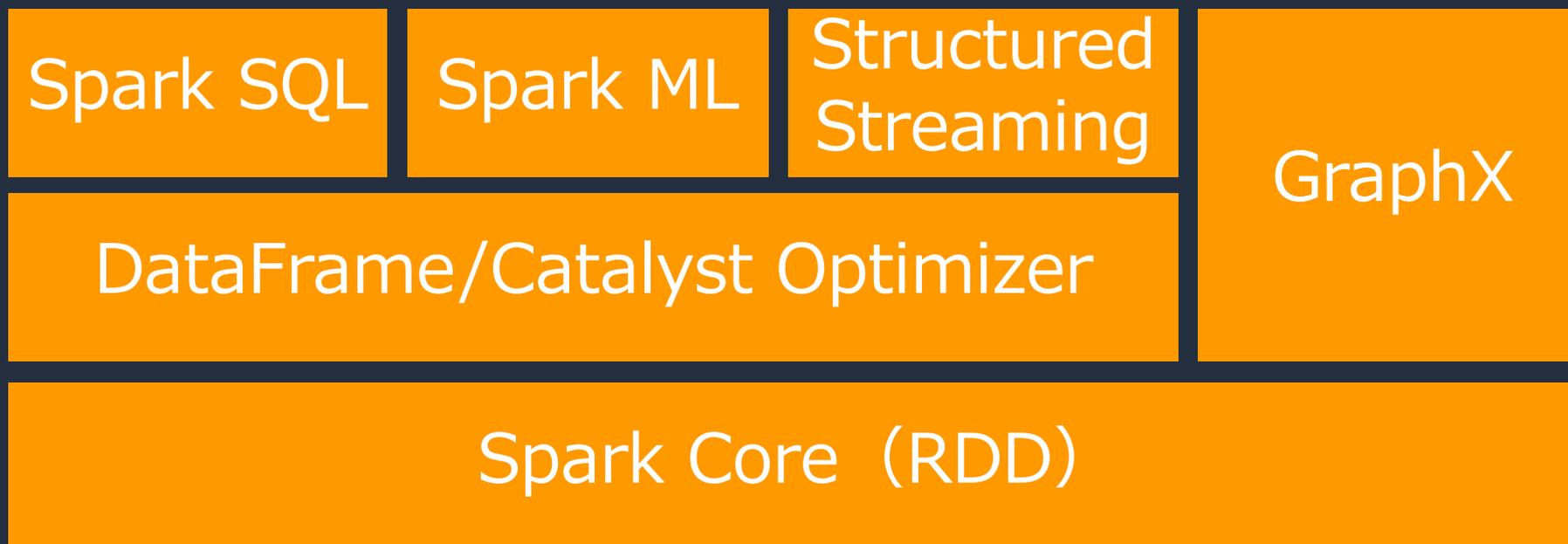


データの流れ

- 処理中のデータはRDD(Resilient Distributed Dataset)と呼ばれる分散コレクションとして定義される
- RDDは1つ以上のパーティションによって成り立っている
- 1つのパーティションは1つの「タスク」で処理される
- コードはDataFrameやDatasetといった型付けのテーブルデータとして扱うことのできるインタフェースを通して記述する事が多い



Apache Sparkのコンポーネント



RDDとDataFrame

- 両方とも1つのリスト/テーブルに対する処理のように処理を記述できるが、実際のデータは複数サーバ上に分散配置される
- DataFrameはRDDの高レベルAPIであり、DataFrameで記述された処理は内部的にはRDDとして実行される

RDDのデータ構造のイメージ

```
[  
  [1, Bob, 24],  
  [2, Alice, 48],  
  [3, Ken, 10],  
  ...  
]
```

DataFrameのデータ構造のイメージ

| col1 | col2 | col3 |
|------|-------|------|
| 1 | Bob | 24 |
| 2 | Alice | 48 |
| 3 | Ken | 10 |
| ... | ... | ... |

処理は「遅延評価」で行われる

- Sparkの処理は、「変換」と「アクション」がある
- 「アクション」が実行されたときにそれに必要な前段の処理が全て行われる
- 「アクション」によって実行される一連の処理のまとまりを「ジョブ」という
- ここでいう「ジョブ」はAWS Glueのジョブとは異なるので注意

```
>>> df1 = spark.read.csv(...)
>>> df2 = spark.read.json(...)
>>> df3 = df1.filter(...)
>>> df4 = spark.read.csv(...)
>>> df5 = df2.join(df3, ...)
>>> df5.count()
```

ここまではコマンドを実行しても
実際に処理は行われない

ここではじめてそれまでの処理が
実行される

アクション

df4の処理はdf5.count()の依存関係にないので、このアクションでは
実行されない

変換とアクションの例

変換：データの生成・加工処理

- select()
 - 列の選択
- read
 - データの読み込み
- filter()
 - データのフィルタリング
- groupBy()
 - グループごとの集計
- sort()
 - データのソート

アクション：処理結果の出力

- count()
 - レコード数のカウント
- write
 - ファイルシステムへの書き出し
- collect()
 - すべてのデータをDriverノード上に集める
- show()
 - データのサンプルの表示
- describe()
 - データの統計情報の表示

Spark アプリケーション

- Sparkアプリケーションは複数のジョブからなる

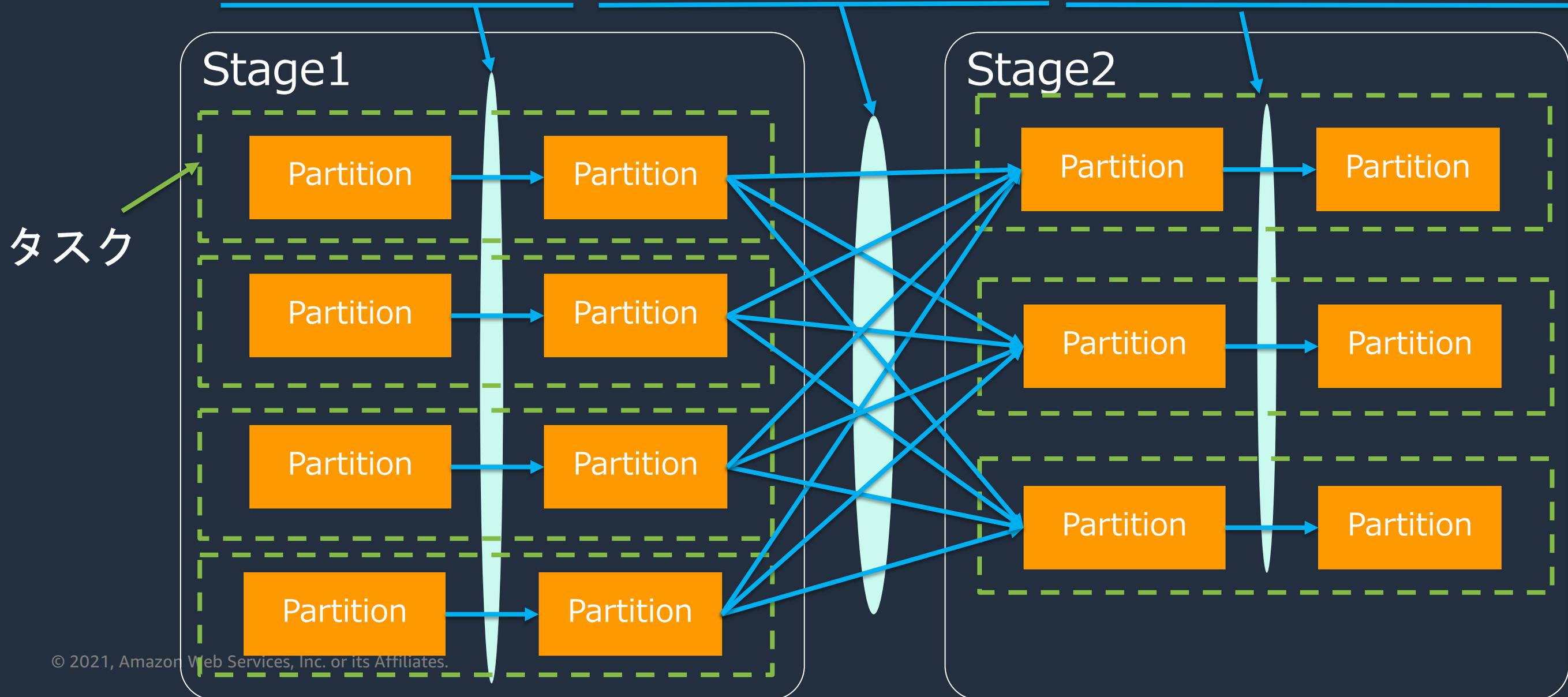
1つのGlueContext（もしくはSparkContext）で実行される処理の一連をアプリケーションという

```
glueContext = GlueContext(SparkContext.getOrCreate(conf))
spark = glueContext.spark_session
df1 = spark.read.json(...)
df1.show() # job1
df1.filter(df1.col1='a').write.parquet(...) # job2
df1.filter(df1.col2='b').write.parquet(...) # job3
```

シャッフル（タスク間のデータのやりとり）とステージ

- シャッフルを区切りとしてステージが分かれる
- 1つのステージでは複数のタスクが並行処理される

```
df2 = df1.filter("price">500).groupBy("item").sum().withColumn("bargain", price*0.8)
```



シャッフルを伴う処理と伴わない処理

シャッフルが行われない例

```
df2 = df1.filter(price>500)
```

シャッフルが行われる例

```
df2 = df1.groupBy('item').sum()
```

df1

| item | price |
|------|-------|
| ハラミ | 1300 |
| カルビ | 200 |
| チキン | 700 |
| ホルモン | 400 |

df2

| item | price |
|------|-------|
| ハラミ | 1300 |
| チキン | 700 |

パーティション

df1

| item | num |
|------|-----|
| ハラミ | 2 |
| カルビ | 3 |
| ハラミ | 4 |
| カルビ | 5 |

df2

| item | num |
|------|-----|
| ハラミ | 6 |
| カルビ | 9 |

シャッフル（タスク間でのデータのやりとり）を伴う処理と伴わない処理

シャッフルを伴わない処理

- read
- filter()
- withColumn()
- UDF
- coalesce()

シャッフルを伴う処理

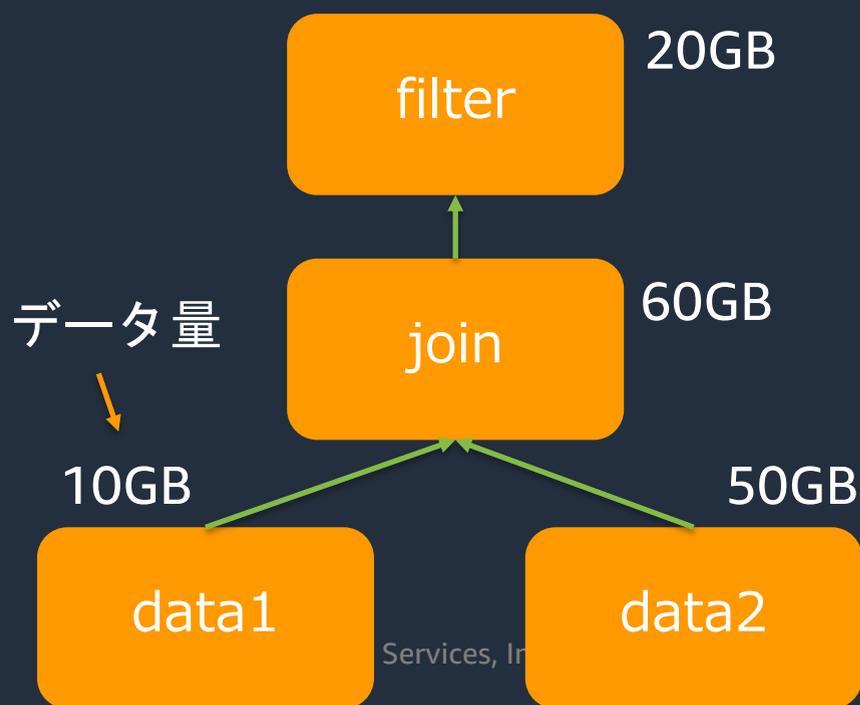
- join()
- groupBy()
- orderBy()
- repartition()

Catalyst Query Optimizerによる最適化

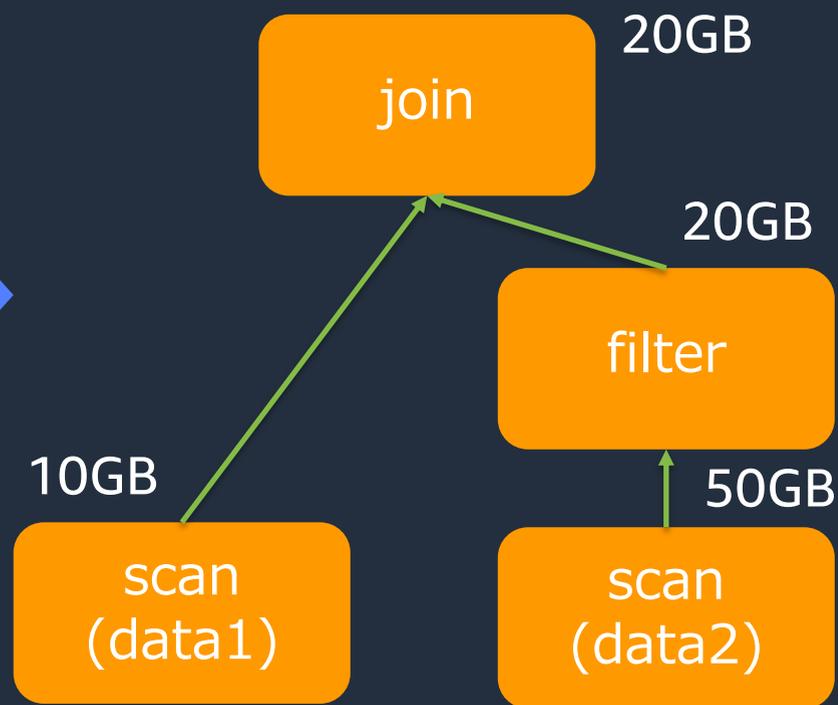
- DataFrameで記述された処理はオプティマイザによって最適化されたRDDに変換されて実行される

```
df1 = spark.read.csv('s3://path/to/data1')
df2 = spark.read.parquet('s3://path/to/data2')
df3 = df1.join(df2, df1.col1 == df2.col1)
df4 = df3.filter(df3.col2 == 'meat').select(df3.col3, df3.col5)
df4.show()
```

Logical Plan

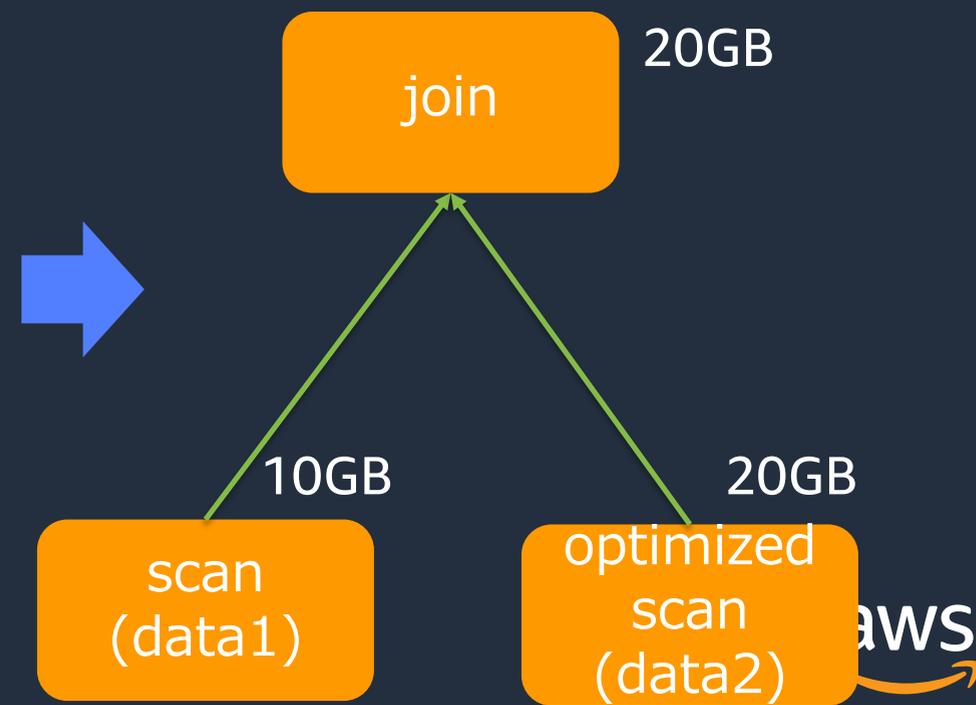


Physical Plan



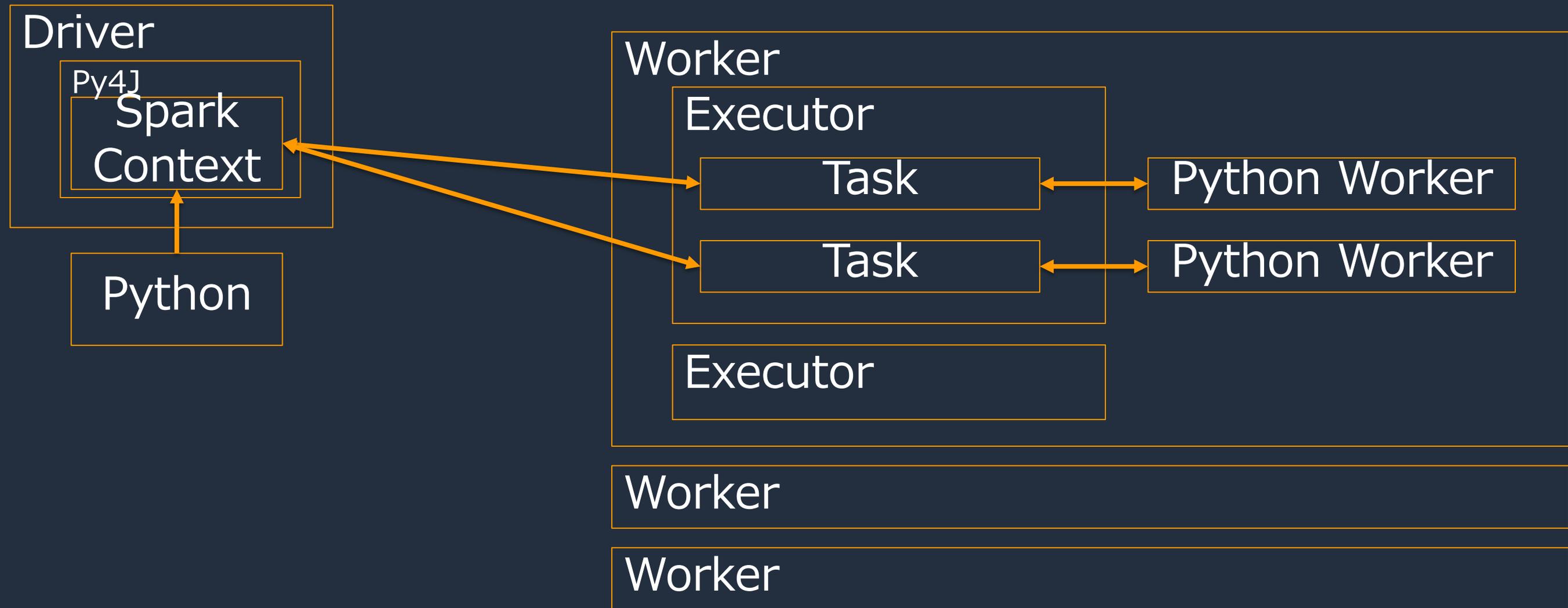
Physical Plan

(with storage side optimization)



PySparkのアーキテクチャ

- PySpark DataFrameで記述された処理はJavaのコードに変換されて実行される
- UDFの内容はタスクごとにPythonワーカーが起動してPythonで実行される

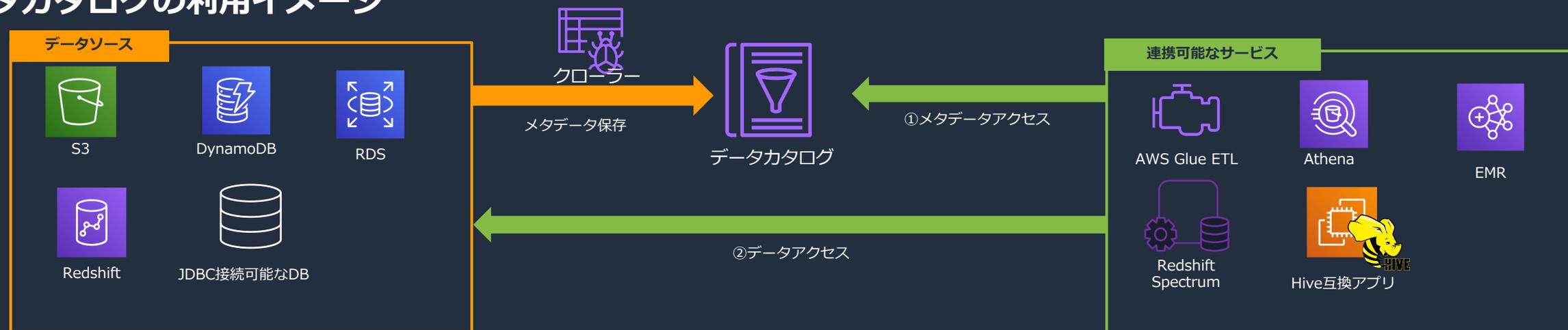


AWS Glue固有の機能のご紹介

データカタログ

- AWS GlueやAmazon Athena、Amazon Redshift Spectrum、Amazon EMRなどからS3やデータベースといったデータソース上のデータにアクセスするときに必要なメタデータ（テーブル名、列名、S3のパスなど）をもっている
- データカタログにメタデータを作成するにはクローラー、AWS GlueのAPI、DDL(Amazon Athena/Amazon EMR/Amazon Redshift Spectrum)の3つの方法が利用可能。
- データソースとして、Amazon DynamoDB、Amazon S3、Amazon Redshift、Amazon RDS、JDBC接続可能なDB、Amazon Kinesis、HDFSなどが指定可能
- メタストアデータベースの管理が不要の為、運用負荷を低減できる

データカタログの利用イメージ



DynamicFrame

生データのETL特有の複雑さを吸収するためのAWS Glueの機能

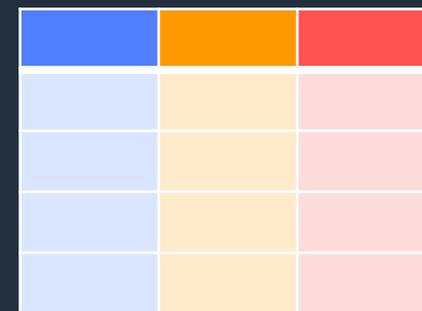
- コンポーネントとしてはDataFrameと同階層に位置し、相互に変換して利用することができる (fromDF・toDF関数)
- 複数の型の可能性を残して、後で決定できるようにする (Choice型)
- DynamicFrameはデータ全体を表し、DynamicRecordはデータ1行を指す

DynamicFrameの位置付け



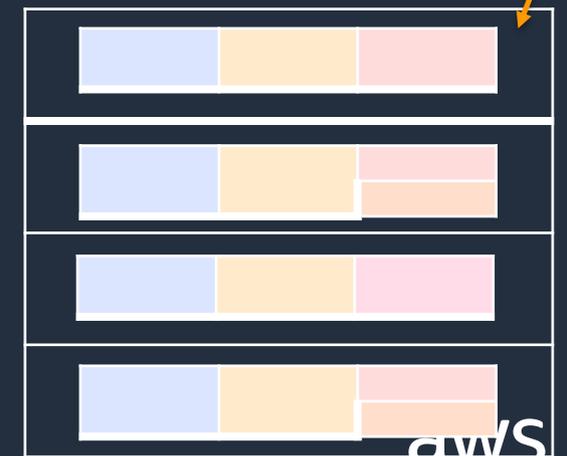
データ構造イメージ

DataFrame



DynamicFrame

record



Choice型

DynamicFrameの列で複数の型を発見した場合に両方の型を持つことができる

- ResolveChoiceメソッドで型を解決することが可能

choice型のデータ構造例

```
root
|-- uuid: string
|-- device id: choice
    |-- long
    |-- string
```

DataFrameでは複数の型が存在した場合、処理が中断し、再処理しなければならない可能性がある

device id列はlongとstringの両方のデータを持っている
(例：device idカラムに数字の1234と文字列の"1234"が混同する)

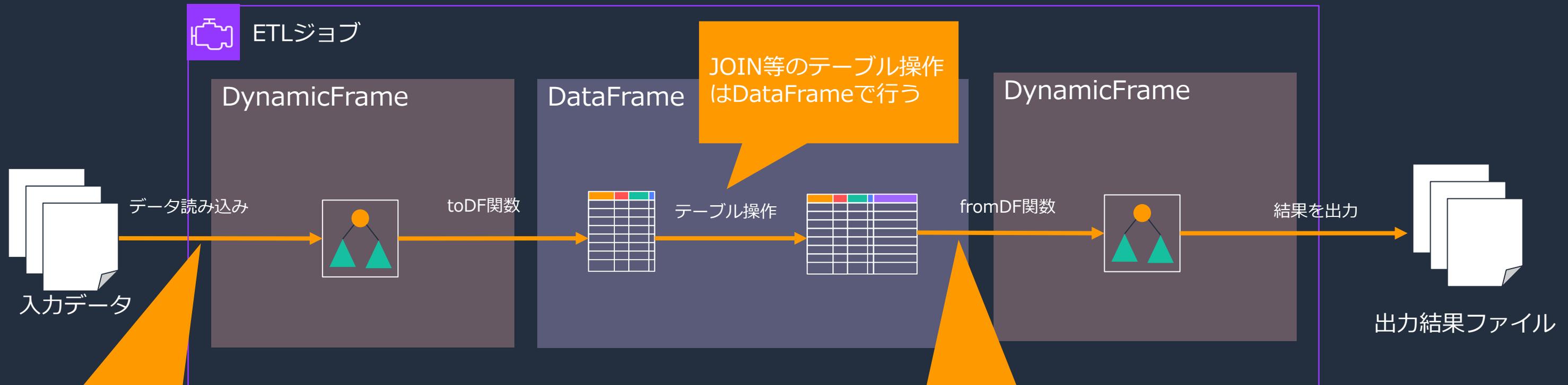
| CoIA | CoIB | CoIC |
|------|------|-----------|
| | | 1 |
| | | 2 |
| | | ... |
| | | 1,000,000 |
| | | "1000001" |
| | | "1000002" |

ResolveChoiceの実行例



DynamicFrameとDataFrameの特性を生かしたETL処理

- DynamicFrameはETL処理に強く、DataFrameはテーブル処理に強い
- データ入出力とそれに伴うETL処理はDynamicFrameで行い、テーブル操作はDataFrameで行う



読み込み時にDynamicFrameを利用することで、AWS Glueカタログを利用したデータ読み込みの最適化、差分データ読み込み、Choice型を利用した半構造化データ加工が可能

toDF・fromDF関数を用いて、DynamicFrameとDataFrameを相互変換（データコピーは行われない。）

ブックマーク機能

定常的なETLを行う場合に差分データのみを処理させる機能

- ファイルのタイムスタンプを利用して、前回ジョブで処理していないデータのみを処理して、重複処理や重複データを防ぐ

```
df = spark.read.parquet('s3://path/to/data')
```

s3://path/to/data



処理済みデータ (読み込まない) 未処理のデータ (読み込む)

AWS Glue ETLのパフォーマンス チューニングの進め方

パフォーマンス・チューニングのサイクル

1. パフォーマンス目標を決める
2. メトリクスを測定する
3. ボトルネックを特定する
4. ボトルネックの影響を軽減する
5. 目標を達成するまで2. から 4. を繰り返す
6. パフォーマンス目標の達成

AWS Glue/Apache Sparkの特性を理解する

- 分散処理
 - 「シャッフル」や「データの偏り」など単一プロセスによるアプリケーションにはないチューニングパターンがある
- 遅延評価
 - Sparkの処理は遅延評価なので、エラーが発生した直前に実行されたAPIではなくその前段に記述されたAPIでエラーとなっていることがある
- オプティマイザによる最適化の影響
 - Sparkの処理は、内部で最適化が行われるので、Spark UIで確認できる実際の処理が書いたスクリプトのどの部分にあたるのかをひもづけるのが難しい場合がある。複数のメトリクスを確認して原因を探る必要がある

AWS GlueにおけるSparkパラメータ

- 本来Sparkでは、ジョブ実行時のパラメータによるチューニング手段があるが、AWS Glueはサーバレスサービスであり、Sparkのパラメータを見る前にまずはAWS Glueのベストプラクティスに則ってチューニングを行う。
- Sparkパラメータの値を変更する場合は十分にテストする。

見るべきメトリクス

- Spark UI (Spark History Server)
 - Sparkの処理の詳細を確認できる
- ExecutorとDriverのログ
 - ExecutorとDriverのstdout/stderrログを確認できる
- AWS Glue ジョブメトリクス
 - 各ExecutorやDriverノードのCPU/メモリなどのリソース使用状況を確認できる
- SparkのAPIで得られる統計情報
 - データのサンプルや統計値を確認できる

チューニングを行うためのジョブ設定

Add job ✕

Configure the job properties

- Job properties
- Data source
- Transform type
- Data target
- Schema

Name

IAM role ⓘ

Choose an IAM role ▼ ↺

Ensure that this role has permission to your Amazon S3 sources, targets, temporary directory, scripts, and any libraries used by the job. [Create IAM role.](#)

Type

Spark ▼

Glue version

Spark 2.4, Python 3 with improved job startup times (Glue Version 2.0) ▼

This job runs

A proposed script generated by AWS Glue ⓘ

An existing script that you provide

A new script to be authored by you

Script file name

S3 path where the script is stored

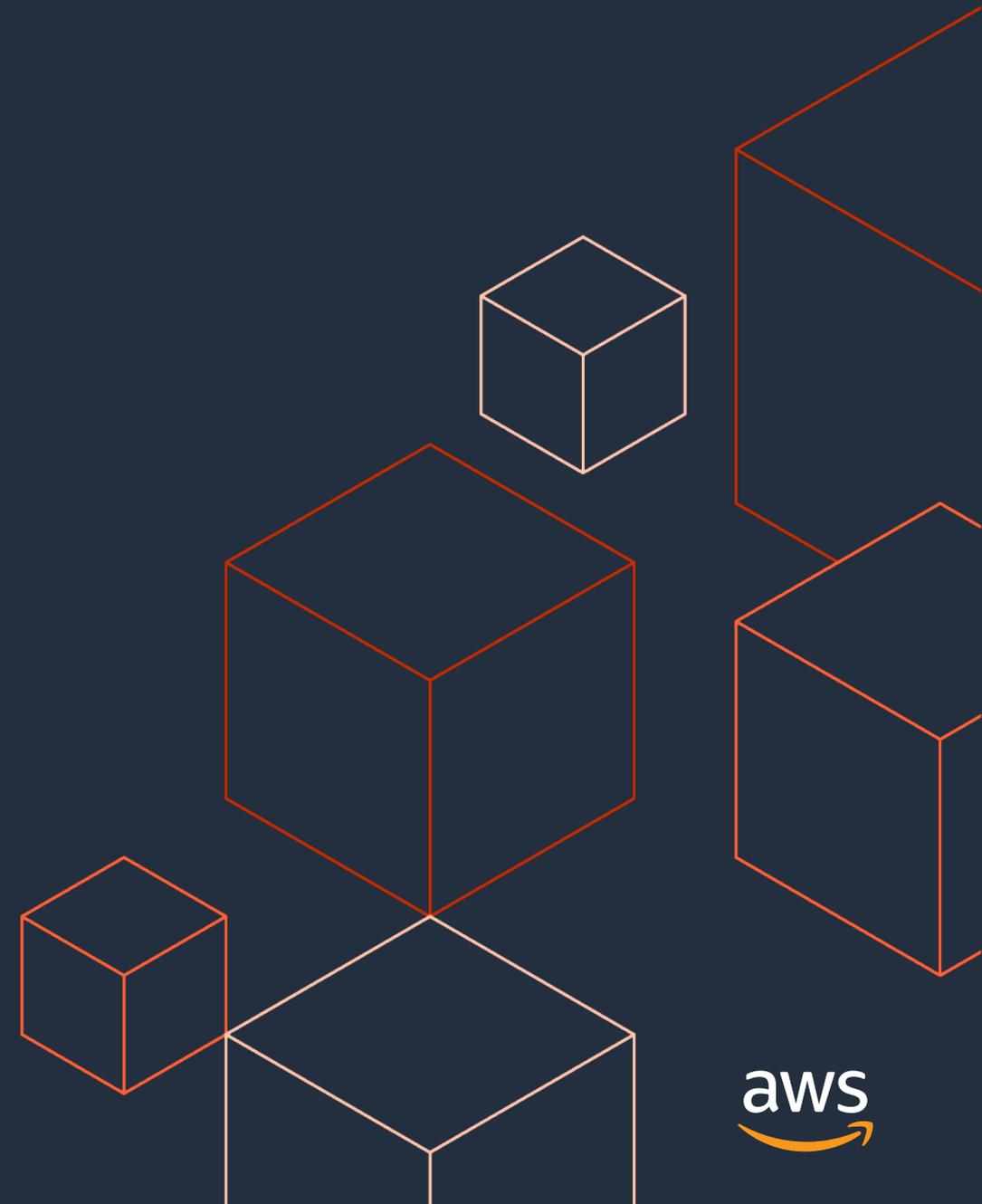
Temporary directory ⓘ

▶ Advanced properties

- ▼ Monitoring options
 - Job metrics ⓘ
 - Continuous logging
 - Spark UI ⓘ

チューニングに必要なログを取得するためには、Add jobの画面でMonitoring Optionsを利用するようにチェックを入れる必要がある

Spark UI



Spark History Server

Add Jobの画面でSpark UIを有効化したときに指定したS3パスにログが出力される
実行済みのアプリケーションのログはSpark History Serverを起動することで
可視化できる

Spark History Serverの起動方法はいくつかある。

- AWS CloudFormationを利用する
- dockerでローカルPCやEC2上で起動する
- ローカルPCやEC2上にApache SparkをダウンロードしてSpark History Serverを起動する
- EMRを利用する

To start the Spark history server and view the Spark UI using AWS CloudFormation

1. Choose one of the **Launch Stack** buttons in the following table. This launches the stack on the AWS CloudFormation console.

| Region | View | Launch |
|--------------------------|----------------------|------------------------------|
| US East (Ohio) | View | Launch Stack |
| US East (N. Virginia) | View | Launch Stack |
| US West (N. California) | View | Launch Stack |
| US West (Oregon) | View | Launch Stack |
| Asia Pacific (Hong Kong) | View | Launch Stack |
| Asia Pacific (Mumbai) | View | Launch Stack |
| Asia Pacific (Seoul) | View | Launch Stack |
| Asia Pacific (Singapore) | View | Launch Stack |
| Asia Pacific (Sydney) | View | Launch Stack |

<https://docs.aws.amazon.com/glue/latest/dg/monitor-spark-ui-history.html>

© 2021, Amazon Web Services, Inc. or its Affiliates.



Dockerで起動する例

- Dockerコンテナが起動したら、ブラウザで <http://localhost:18080> にアクセス

Spark UIのログの格納先を指定

```
$ git clone aws-samples/aws-glue-samples
$ cd aws-glue-samples/utilities/Spark_UI/glue-1_0-2_0
$ docker build -t glue/sparkui:latest .
$ docker run -itd -e SPARK_HISTORY_OPTS="$SPARK_HISTORY_OPTS -Dspark.history.fs.logDirectory=s3a://path_to_eventlog
-Dspark.hadoop.fs.s3a.access.key=AWS_ACCESS_KEY_ID -Dspark.hadoop.fs.s3a.secret.key=AWS_SECRET_ACCESS_KEY" -p
18080:18080 glue/sparkui:latest "/opt/spark/bin/spark-class org.apache.spark.deploy.history.HistoryServer"
```

Spark History Server



Event log directory: s3a://glueworkshop-chie/sparkeventlogs

There are 990 event log(s) currently being processed which may result in additional applications getting listed on this page. Refresh the page to view updates.

Client local time zone: Asia/Tokyo

クリックするとそのアプリケーションの詳細へ

アプリケーションの実行時間を確認

| App ID | App Name | Started | Completed | Duration | Spark User | Last Updated | Event Log |
|--|-----------------|---------------------|---------------------|----------|------------|---------------------|--------------------------|
| application_1622857946319_0011 | livy-session-10 | 2021-06-15 12:05:21 | 2021-06-15 13:05:49 | 1.0 h | livy | 2021-06-15 13:06:11 | Download |
| application_1622857946319_0010 | livy-session-9 | 2021-06-09 21:28:16 | 2021-06-09 22:28:48 | 1.0 h | livy | 2021-06-09 22:29:11 | Download |
| application_1622857946319_0009 | livy-session-8 | 2021-06-09 21:19:27 | 2021-06-09 21:27:56 | 8.5 min | livy | 2021-06-09 21:28:11 | Download |
| application_1622857946319_0008 | livy-session-7 | 2021-06-09 21:14:36 | 2021-06-09 21:18:27 | 3.9 min | livy | 2021-06-09 21:18:41 | Download |
| application_1622857946319_0007 | livy-session-6 | 2021-06-09 20:47:25 | 2021-06-09 21:14:15 | 27 min | livy | 2021-06-09 21:14:41 | Download |
| application_1622857946319_0006 | livy-session-5 | 2021-06-09 20:31:32 | 2021-06-09 20:47:05 | 16 min | livy | 2021-06-09 20:47:11 | Download |
| application_1622857946319_0005 | livy-session-4 | 2021-06-09 20:04:45 | 2021-06-09 20:20:06 | 15 min | livy | 2021-06-09 20:20:11 | Download |
| application_1622857946319_0004 | livy-session-3 | 2021-06-09 20:02:11 | 2021-06-09 20:03:10 | 59 s | livy | 2021-06-09 20:03:11 | Download |
| application_1622857946319_0003 | livy-session-2 | 2021-06-09 15:46:56 | 2021-06-09 15:49:01 | 2.1 min | livy | 2021-06-09 15:49:11 | Download |
| application_1622857946319_0002 | livy-session-1 | 2021-06-09 13:58:35 | 2021-06-09 15:46:36 | 1.8 h | livy | 2021-06-09 15:46:41 | Download |

Showing 1 to 10 of 10 entries
[Show incomplete applications](#)



アプリケーションで実行されたジョブ一覧

APACHE Spark 2.4.3 Jobs Stages Storage Environment Executors SQL livy-session-1 application UI

Spark Jobs (?)

User: livy
Total Uptime: 1.8 h
Scheduling Mode: FIFO
Completed Jobs: 20
Failed Jobs: 1

成功したジョブ一覧

時間のかかったジョブを確認

▶ Event Timeline
▼ Completed Jobs (20)

| Job Id (Job Group) ▼ | Description | Submitted | Duration | Stages: Succeeded/Total | Tasks (for all stages): Succeeded/Total |
|----------------------|---|---------------------|----------|-------------------------|---|
| 20 (14) | Job group for statement 14 sql at NativeMethodAccessorImpl.java:0 | 2021/06/09 06:41:34 | 0.7 s | 1/1 | 1/1 |
| 19 (14) | Listing leaf files and directories for 39 paths: s3://chie-us-east-2/tmp/parquet121/product_category=Digital_Music_Purchase, ... sql at NativeMethodAccessorImpl.java:0 | 2021/06/09 06:41:29 | 4 s | 1/1 | 39/39 |
| 18 (13) | Job group for statement 13 parquet at NativeMethodAccessorImpl.java:0 | 2021/06/09 06:36:27 | 5.0 min | 1/1 | 5/5 |
| 17 (13) | Job group for statement 13 csv at NativeMethodAccessorImpl.java:0 | 2021/06/09 06:34:32 | 1.9 min | 1/1 | 5/5 |
| 16 (13) | Job group for statement 13 csv at NativeMethodAccessorImpl.java:0 | 2021/06/09 06:34:32 | 0.2 s | 1/1 | 1/1 |
| 15 (12) | Job group for statement 12 parquet at NativeMethodAccessorImpl.java:0 | 2021/06/09 06:28:40 | 5.3 min | 1/1 | 5/5 |
| 14 (12) | Job group for statement 12 csv at NativeMethodAccessorImpl.java:0 | 2021/06/09 06:27:02 | 1.6 min | 1/1 | 5/5 |
| 13 (12) | Job group for statement 12 csv at NativeMethodAccessorImpl.java:0 | 2021/06/09 06:26:54 | 8 s | 1/1 | 1/1 |
| 12 (10) | Job group for statement 10 parquet at NativeMethodAccessorImpl.java:0 | 2021/06/09 06:21:43 | 3.3 min | 1/1 | 5/5 |
| 11 (10) | Job group for statement 10 csv at NativeMethodAccessorImpl.java:0 | 2021/06/09 06:20:03 | 1.6 min | 1/1 | 5/5 |
| 10 (10) | Job group for statement 10 csv at NativeMethodAccessorImpl.java:0 | 2021/06/09 06:19:54 | 9 s | 1/1 | 1/1 |
| 9 (9) | | 2021/06/09 05:50:48 | 38 s | 2/2 | 3/3 |

失敗したジョブ一覧

▼ Failed Jobs (1)

| Job Id (Job Group) ▼ | Description | Submitted | Duration | Stages: Succeeded/Total | Tasks (for all stages): Succeeded/Total |
|----------------------|---|---------------------|----------|-------------------------|---|
| 4 (4) | Job group for statement 4 parquet at NativeMethodAccessorImpl.java:0 | 2021/06/09 05:14:19 | 4.1 min | 0/1 (1 failed) | 4/5 (1 killed: Stage cancelled) |

ジョブの中身の確認

Spark 2.4.3 Jobs Stages Storage Environment Executors SQL livy-session-25 application UI

Details for Job 1

Status: FAILED
Job Group: 18
Completed Stages: 1
Skipped Stages: 1
Failed Stages: 1

Event Timeline
DAG Visualization

Stage 1 (Completed): newAPIHadoopRDD, map, mapPartitionsWithIndex, filter, map, Scan ExistingRDD, WholeStageCodegen, Exchange

Stage 2 (Completed): newAPIHadoopRDD, map, mapPartitionsWithIndex, filter, map, Scan ExistingRDD, WholeStageCodegen, Exchange

Stage 3 (skipped): Exchange, WholeStageCodegen, InMemoryTableScan, WholeStageCodegen, mapPartitionsInternal, map

Failしていたり時間のかかっている Stageを確認

Completed Stages (1)

| Stage Id | Description | Duration | Tasks: Succeeded/Total | Input | Output | Shuffle Read | Shuffle Write |
|----------|---|----------|------------------------|--------|--------|--------------|---------------|
| 2 | Job group for statement 18 showString at NativeMethodAccessorImpl.java:0 | 1.8 min | 5/5 | 2.1 GB | | | 11.6 KB |

Skipped Stages (1)

| Stage Id | Description | Submitted | Duration | Tasks: Succeeded/Total | Input | Output | Shuffle Read | Shuffle Write |
|----------|---|-----------|----------|------------------------|-------|--------|--------------|---------------|
| 3 | showString at NativeMethodAccessorImpl.java:0 | Unknown | Unknown | 0/1 | | | | |

Failed Stages (1)

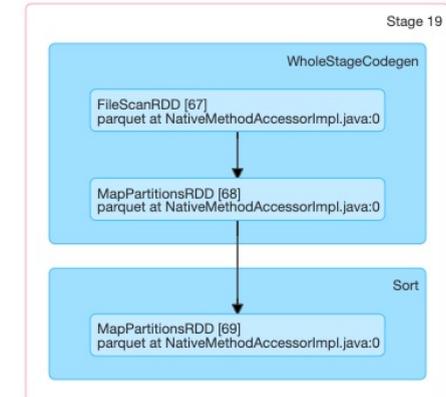
| Stage Id | Description | Submitted | Duration | Tasks: Succeeded/Total | Input | Output | Shuffle Read | Shuffle Write | Failure Reason |
|----------|---|---------------------|----------|---------------------------------|-----------|--------|--------------|---------------|--|
| 1 | Job group for statement 18 showString at NativeMethodAccessorImpl.java:0 | 2021/05/30 12:10:39 | 1.8 min | 4/5 (1 killed: Stage cancelled) | 1864.7 MB | | | 1304.6 MB | Job 1 cancelled part of cancelled job group 18 |

Stageの中身の確認

Details for Stage 19 (Attempt 0)

Total Time Across All Tasks: 10 min
Locality Level Summary: Process local: 5
Input Size / Records: 2.1 GB / 9834292
Output: 2.8 GB / 9834292
Shuffle Spill (Memory): 1888.0 MB
Shuffle Spill (Disk): 1141.8 MB

▼ DAG Visualization

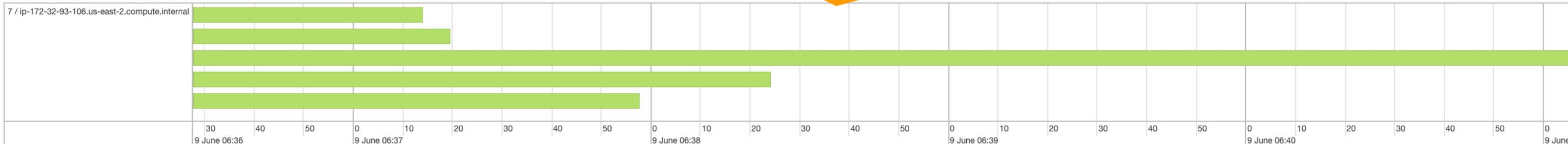


▶ Show Additional Metrics

▼ Event Timeline

Enable zooming

■ Scheduler Delay ■ Executor Computing Time ■ Getting Result Time
■ Task Deserialization Time ■ Shuffle Write Time
■ Shuffle Read Time ■ Result Serialization Time



Summary Metrics for 5 Completed Tasks

| Metric | Min | 25th percentile | Median | 75th percentile | Max |
|----------|------|-----------------|---------|-----------------|---------|
| Duration | 46 s | 52 s | 1.5 min | 1.9 min | 5.0 min |

Stageでのデータサイズが確認できる

ラインの長さに差がある場合は処理の偏りが発生して分散処理ができていない



Stageの中身の確認 (続き)



上記のEvent Timelineの他に Summary MetricsやTasksでも処理の偏りの状況が確認できる

ExecutorごとのTask Time

何に時間がかかっているかの詳細が確認できる

DiskへのSpillがある場合はよりメモリが大きいWorkerノードを選ぶと解消する

Summary Metrics for 5 Completed Tasks

| Metric | Min | 25th percentile | 50th percentile | 75th percentile | Max |
|---------------------------|---------------------|-------------------|--------------------|--------------------|---------------------|
| Duration | 46 s | 52 s | 1.5 min | 1.9 min | 5.0 min |
| Scheduler Delay | 4 ms | 9 ms | 10 ms | 11 ms | 11 ms |
| Task Deserialization Time | 48 ms | 49 ms | 54 ms | 59 ms | 61 ms |
| GC Time | 10 s | 11 s | 16 s | 19 s | 29 s |
| Result Serialization Time | 0 ms | 0 ms | 0 ms | 0 ms | 0 ms |
| Getting Result Time | 0 ms | 0 ms | 0 ms | 0 ms | 0 ms |
| Peak Execution Memory | 0.0 B | 0.0 B | 0.0 B | 0.0 B | 0.0 B |
| Input Size / Records | 230.7 MB / 679121 | 322.9 MB / 679121 | 333.2 MB / 1707494 | 433.3 MB / 1707494 | 1399.0 MB / 6931166 |
| Output Size / Records | 1851.9 MB / 6931166 | 1888.0 MB | 0.0 B | 0.0 B | 0.0 B |
| Shuffle spill (memory) | 0.0 B | 0.0 B | 0.0 B | 0.0 B | 1888.0 MB |
| Shuffle spill (disk) | 0.0 B | 0.0 B | 0.0 B | 0.0 B | 0.0 B |

Aggregated Metrics by Executor

| Executor ID | Address | Task Time | Total Tasks | Failed Tasks | Killed Tasks | Succeeded Tasks |
|-------------|---|-----------|-------------|--------------|--------------|-----------------|
| 7 | ip-172-32-93-106.us-east-2.compute.internal:39651 | 10 min | 5 | 0 | 0 | 5 |

Tasks (5)

| Index | ID | Attempt | Status | Locality Level | Executor ID | Host | Launch Time | Duration | Scheduler Delay | Task Deserialization Time | GC Time | Result Serialization Time | Getting Result Time | Peak Execution Memory | Input Size / Records | Output Size / Records | Shuffle Spill (Memory) | Shuffle Spill (Disk) | Errors |
|-------|----|---------|---------|----------------|-------------|---|---------------------|----------|-----------------|---------------------------|---------|---------------------------|---------------------|-----------------------|----------------------|-----------------------|------------------------|----------------------|--------|
| 0 | 89 | 0 | SUCCESS | PROCESS_LOCAL | 7 | ip-172-32-93-106.us-east-2.compute.internal | 2021/06/09 06:36:27 | 5.0 min | 4 ms | 48 ms | 29 s | 0 ms | 0 ms | 0.0 B | 1399.0 MB / 6931166 | 1851.9 MB / 6931166 | 1888.0 MB | 1141.8 MB | |
| 1 | 90 | 0 | SUCCESS | PROCESS_LOCAL | 7 | ip-172-32-93-106.us-east-2.compute.internal | 2021/06/09 06:36:27 | 1.9 min | 9 ms | 61 ms | 19 s | 0 ms | 0 ms | 0.0 B | 333.2 MB / 1707494 | 433.3 MB / 1707494 | 0.0 B | 0.0 B | |
| 2 | 91 | 0 | SUCCESS | PROCESS_LOCAL | 7 | ip-172-32-93-106.us-east-2.compute.internal | 2021/06/09 06:36:27 | 1.5 min | 11 ms | 54 ms | 16 s | 0 ms | 0 ms | 0.0 B | 0.0 B | 0.0 B | 0.0 B | | |
| 3 | 92 | 0 | SUCCESS | PROCESS_LOCAL | 7 | ip-172-32-93-106.us-east-2.compute.internal | 2021/06/09 06:36:27 | 52 s | 10 ms | 59 ms | 11 s | 0 ms | 0 ms | 0.0 B | 0.0 B | 0.0 B | 0.0 B | | |
| 4 | 93 | 0 | SUCCESS | PROCESS_LOCAL | 7 | ip-172-32-93-106.us-east-2.compute.internal | 2021/06/09 06:36:27 | 46 s | 11 ms | 49 ms | 10 s | 0 ms | 0 ms | 0.0 B | 0.0 B | 0.0 B | 0.0 B | | |

Failしているタスクのdetailを見る

detailsをクリックして
詳細を確認

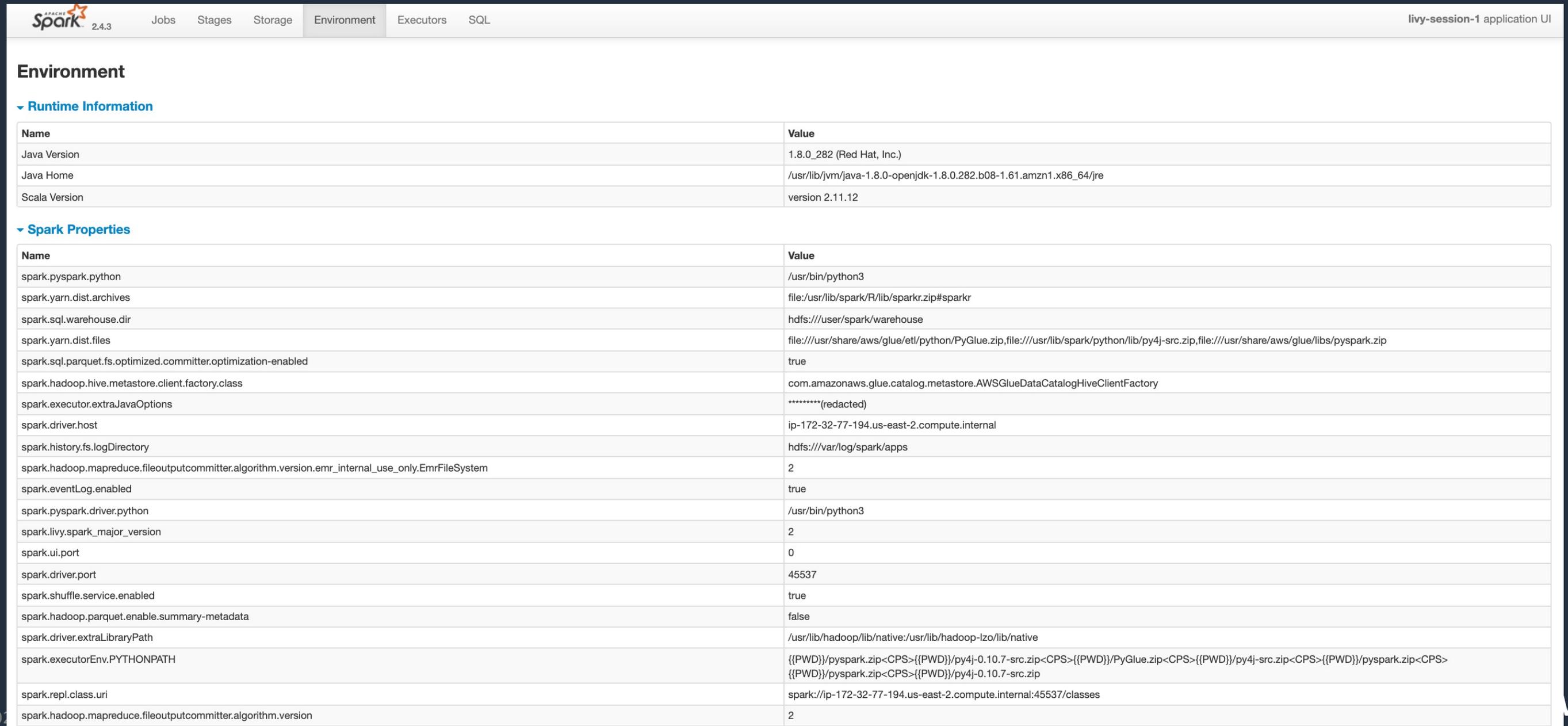
Tasks (86)

| Index ▲ | ID | Attempt | Status | Locality Level | Executor ID | Host | Launch Time | Duration | Scheduler Delay | Task Deserialization Time | GC Time | Result Serialization Time | Getting Result Time | Peak Execution Memory | Input Size / Records | Write Time | Shuffle Write Size / Records | Errors |
|---------|-----|---------|---------|----------------|-------------|--|---------------------|----------|-----------------|---------------------------|---------|---------------------------|---------------------|-----------------------|----------------------|------------|------------------------------|---|
| 0 | 149 | 0 | FAILED | PROCESS_LOCAL | 9 | ip-172-32-30-47.us-east-2.compute.internal | 2021/05/26 09:50:45 | 0 ms | 6.7 min | 0 ms | | 0 ms | 0 ms | 0.0 B | 0.0 B / 0 | | 0.0 B / 0 | ExecutorLostFailure (executor 9 exited caused by one of the running tasks) Reason: Container from a bad node: container_1621981678428_0004_01_000013 on host: ip-172-32-30-47.us-east-2.compute.internal. Exit status: 137. Diagnostics: Container killed on request. Exit code is 137 +details ExecutorLostFailure (executor 9 exited caused by one of the running tasks) Reason: Container from a bad node: container_1621981678428_0004_01_000013 on host: ip-172-32-30-47.us-east-2.compute.internal. Exit status: 137. Diagnostics: Container killed on request. Exit code is 137 Container exited with a non-zero exit code 137 Killed by external signal |
| 0 | 216 | 1 | SUCCESS | PROCESS_LOCAL | 17 | ip-172-32-5-152.us-east-2.compute.internal | 2021/05/26 09:57:30 | 16 min | 7 ms | | 1.0 min | 0 ms | 0 ms | 0.0 B | 2.6 GB / 10319090 | 5 ms | 2.4 KB / 41 | |
| 1 | 150 | 0 | FAILED | PROCESS_LOCAL | 4 | ip-172-32-30-47.us-east-2.compute.internal | 2021/05/26 09:50:45 | 0 ms | 5.6 min | 0 ms | | 0 ms | 0 ms | 0.0 B | 0.0 B / 0 | | 0.0 B / 0 | ExecutorLostFailure (executor 4 exited caused by one of the running tasks) Reason: Container from a bad node: container_1621981678428_0004_01_000006 on host: ip-172-32-30-47.us-east-2.compute.internal. Exit status: 137. Diagnostics: Container killed on request. Exit code is 137 +details |
| 1 | 209 | 1 | SUCCESS | PROCESS_LOCAL | 16 | ip-172-32-20-51.us-east-2.compute.internal | 2021/05/26 09:56:21 | 15 min | 18 ms | 4 ms | | 0 ms | 0 ms | 0.0 B | 0.0 B / 0 | | 0.0 B / 0 | |
| 2 | 151 | 0 | FAILED | PROCESS_LOCAL | 7 | ip-172-32-18-1.us-east-2.compute.internal | 2021/05/26 09:50:45 | 0 ms | 4.5 min | 0 ms | | 0 ms | 0 ms | 0.0 B | 0.0 B / 0 | | 0.0 B / 0 | ExecutorLostFailure (executor 7 exited caused by one of the running tasks) Reason: Container from a bad node: container_1621981678428_0004_01_000009 on host: ip-172-32-18-1.us-east-2.compute.internal. Exit status: 137. Diagnostics: Container killed on request. Exit code is 137 +details |
| 2 | 202 | 1 | SUCCESS | PROCESS_LOCAL | 16 | ip-172-32-20-51.us-east-2.compute.internal | 2021/05/26 09:55:18 | 17 min | 5 ms | 5 ms | | 0 ms | 0 ms | 0.0 B | 0.0 B / 0 | | 0.0 B / 0 | |
| 3 | 152 | 0 | FAILED | PROCESS_LOCAL | 9 | ip-172-32-30-47.us-east-2.compute.internal | 2021/05/26 09:50:45 | 0 ms | 6.7 min | 0 ms | | 0 ms | 0 ms | 0.0 B | 0.0 B / 0 | | 0.0 B / 0 | ExecutorLostFailure (executor 9 exited caused by one of the running tasks) Reason: Container from a bad node: container_1621981678428_0004_01_000013 on host: ip-172-32-30-47.us-east-2.compute.internal. Exit status: 137. Diagnostics: Container killed on request. Exit code is 137 +details |
| 3 | 215 | 1 | SUCCESS | PROCESS_LOCAL | 17 | ip-172-32-5-152.us-east-2.compute.internal | 2021/05/26 09:57:30 | 14 min | 8 ms | 6 ms | 3.9 min | 0 ms | 0 ms | 0.0 B | 1625.7 MB / 9002021 | 11 ms | 8.9 KB / 197 | |
| 4 | 153 | 0 | FAILED | PROCESS_LOCAL | 4 | ip-172-32-30-47.us-east-2.compute.internal | 2021/05/26 09:50:45 | 0 ms | 5.6 min | 0 ms | | 0 ms | 0 ms | 0.0 B | 0.0 B / 0 | | 0.0 B / 0 | ExecutorLostFailure (executor 4 exited caused by one of the running tasks) Reason: Container from a bad node: container_1621981678428_0004_01_000006 on host: ip-172-32-30-47.us-east-2.compute.internal. Exit status: 137. Diagnostics: Container killed on request. Exit code is 137 +details |
| 4 | 207 | 1 | FAILED | PROCESS_LOCAL | 19 | ip-172-32-6-165.us-east-2.compute.internal | 2021/05/26 09:56:21 | 0 ms | 2.2 min | 0 ms | | 0 ms | 0 ms | 0.0 B | 0.0 B / 0 | | 0.0 B / 0 | ExecutorLostFailure (executor 19 exited caused by one of the running tasks) Reason: Container from a bad node: container_1621981678428_0004_01_000026 on host: ip-172-32-6-165.us-east-2.compute.internal. Exit status: 137. Diagnostics: Container killed on request. Exit code is 137 +details |
| 4 | 220 | 2 | FAILED | PROCESS_LOCAL | 23 | ip-172-32-29-74.us-east-2.compute.internal | 2021/05/26 09:56:21 | 0 ms | 4.1 min | 0 ms | | 0 ms | 0 ms | 0.0 B | 0.0 B / 0 | | 0.0 B / 0 | ExecutorLostFailure (executor 23 exited caused by one of the running tasks) Reason: Container from a bad node: container_1621981678428_0004_01_000031 on host: ip-172-32-29-74.us-east-2.compute.internal. Exit status: 137. Diagnostics: Container killed on request. Exit code is 137 +details |

Failが起きているExecutorのログを見る。
AWS Glue ETLの場合はExecutor IDを確認してCloudWatchのLog groupsから確認する。

Sparkアプリケーション実行時の環境設定の一覧

- Sparkのオプションなどが確認できる



The screenshot displays the 'Environment' section of the Spark application UI. It is divided into two main parts: 'Runtime Information' and 'Spark Properties'.

Runtime Information

| Name | Value |
|---------------|---|
| Java Version | 1.8.0_282 (Red Hat, Inc.) |
| Java Home | /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.282.b08-1.61.amzn1.x86_64/jre |
| Scala Version | version 2.11.12 |

Spark Properties

| Name | Value |
|--|---|
| spark.pyspark.python | /usr/bin/python3 |
| spark.yarn.dist.archives | file:/usr/lib/spark/R/lib/sparkr.zip#sparkr |
| spark.sql.warehouse.dir | hdfs://user/spark/warehouse |
| spark.yarn.dist.files | file:///usr/share/aws/glue/etl/python/PyGlue.zip,file:///usr/lib/spark/python/lib/py4j-src.zip,file:///usr/share/aws/glue/libs/pyspark.zip |
| spark.sql.parquet.fs.optimized.committer.optimization-enabled | true |
| spark.hadoop.hive.metastore.client.factory.class | com.amazonaws.glue.catalog.metastore.AWSGlueDataCatalogHiveClientFactory |
| spark.executor.extraJavaOptions | ***** (redacted) |
| spark.driver.host | ip-172-32-77-194.us-east-2.compute.internal |
| spark.history.fs.logDirectory | hdfs://var/log/spark/apps |
| spark.hadoop.mapreduce.fileoutputcommitter.algorithm.version.emr_internal_use_only.EmrFileSystem | 2 |
| spark.eventLog.enabled | true |
| spark.pyspark.driver.python | /usr/bin/python3 |
| spark.livy.spark_major_version | 2 |
| spark.ui.port | 0 |
| spark.driver.port | 45537 |
| spark.shuffle.service.enabled | true |
| spark.hadoop.parquet.enable.summary-metadata | false |
| spark.driver.extraLibraryPath | /usr/lib/hadoop/lib/native:/usr/lib/hadoop-lzo/lib/native |
| spark.executorEnv.PYTHONPATH | {{PWD}}/pyspark.zip<CPS>{{PWD}}/py4j-0.10.7-src.zip<CPS>{{PWD}}/PyGlue.zip<CPS>{{PWD}}/py4j-src.zip<CPS>{{PWD}}/pyspark.zip<CPS>{{PWD}}/pyspark.zip<CPS>{{PWD}}/py4j-0.10.7-src.zip |
| spark.repl.class.uri | spark://ip-172-32-77-194.us-east-2.compute.internal:45537/classes |
| spark.hadoop.mapreduce.fileoutputcommitter.algorithm.version | 2 |

DriverとExecutorのノード一覧

| APACHE Spark 2.4.3 | | | | | | | | | | | | | | Jobs | | Stages | | Storage | | Environment | | Executors | | SQL | | livy-session-1 application UI | |
|--|---|-----------------|------------|----------------|--------------|--------------|----------------|--------------|---------------------|-------------|---------------------|---------------|--------------|---------------|--|--------|--|---------|--|-------------|--|-----------|--|-----|--|-------------------------------|--|
| Executors | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Summary | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | RDD Blocks | Storage Memory | Disk Used | Cores | Active Tasks | Failed Tasks | Complete Tasks | Total Tasks | Task Time (GC Time) | Input | Shuffle Read | Shuffle Write | Blacklisted | | | | | | | | | | | | | | |
| Active(1) | 0 | 0.0 B / 6.2 GB | 0.0 B | 0 | 0 | 0 | 0 | 0 | 0 ms (0 ms) | 0.0 B | 0.0 B | 0.0 B | 0 | | | | | | | | | | | | | | |
| Dead(9) | 0 | 0.0 B / 56.1 GB | 0.0 B | 72 | 0 | 0 | 133 | 134 | 53 min (5.5 min) | 18 GB | 3.4 KB | 3.4 KB | 0 | | | | | | | | | | | | | | |
| Total(10) | 0 | 0.0 B / 62.3 GB | 0.0 B | 72 | 0 | 0 | 133 | 134 | 53 min (5.5 min) | 18 GB | 3.4 KB | 3.4 KB | 0 | | | | | | | | | | | | | | |
| Executors | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Show <input type="text" value="20"/> entries Search: <input type="text"/> | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Executor ID | Address | Status | RDD Blocks | Storage Memory | Disk Used | Cores | Active Tasks | Failed Tasks | Complete Tasks | Total Tasks | Task Time (GC Time) | Input | Shuffle Read | Shuffle Write | Logs | | | | | | | | | | | | |
| driver | ip-172-32-77-194.us-east-2.compute.internal:34155 | Active | 0 | 0.0 B / 6.2 GB | 0.0 B | 0 | 0 | 0 | 0 | 0 | 0 ms (0 ms) | 0.0 B | 0.0 B | 0.0 B | stdout stderr | | | | | | | | | | | | |
| 1 | ip-172-32-72-193.us-east-2.compute.internal:41143 | Dead | 0 | 0.0 B / 6.2 GB | 0.0 B | 8 | 0 | 0 | 2 | 2 | 16 s (0.3 s) | 129.9 MB | 0.0 B | 0.0 B | stdout stderr | | | | | | | | | | | | |
| 2 | ip-172-32-89-85.us-east-2.compute.internal:33663 | Dead | 0 | 0.0 B / 6.2 GB | 0.0 B | 8 | 0 | 0 | 43 | 44 | 13 min (1.4 min) | 4.5 GB | 0.0 B | 0.0 B | stdout stderr | | | | | | | | | | | | |
| 3 | ip-172-32-72-193.us-east-2.compute.internal:37557 | Dead | 0 | 0.0 B / 6.2 GB | 0.0 B | 8 | 0 | 0 | 10 | 10 | 45 s (1 s) | 14.1 KB | 0.0 B | 0.0 B | stdout stderr | | | | | | | | | | | | |
| 4 | ip-172-32-94-150.us-east-2.compute.internal:35111 | Dead | 0 | 0.0 B / 6.2 GB | 0.0 B | 8 | 0 | 0 | 2 | 2 | 11 s (0.1 s) | 14.1 KB | 0.0 B | 0.0 B | stdout stderr | | | | | | | | | | | | |
| 5 | ip-172-32-81-68.us-east-2.compute.internal:37323 | Dead | 0 | 0.0 B / 6.2 GB | 0.0 B | 8 | 0 | 0 | 3 | 3 | 44 s (0.7 s) | 32.1 MB | 3.4 KB | 3.4 KB | stdout stderr | | | | | | | | | | | | |
| 6 | ip-172-32-64-174.us-east-2.compute.internal:44055 | Dead | 0 | 0.0 B / 6.2 GB | 0.0 B | 8 | 0 | 0 | 11 | 11 | 9.7 min (38 s) | 4.4 GB | 0.0 B | 0.0 B | stdout stderr | | | | | | | | | | | | |
| 7 | ip-172-32-93-106.us-east-2.compute.internal:39651 | Dead | 0 | 0.0 B / 6.2 GB | 0.0 B | 8 | 0 | 0 | 62 | 62 | 28 min (3.4 min) | 8.9 GB | 0.0 B | 0.0 B | stdout stderr | | | | | | | | | | | | |
| 8 | ip-172-32-95-192.us-east-2.compute.internal:46277 | Dead | 0 | 0.0 B / 6.2 GB | 0.0 B | 8 | 0 | 0 | 0 | 0 | 0 ms (0 ms) | 0.0 B | 0.0 B | 0.0 B | stdout stderr | | | | | | | | | | | | |
| 9 | ip-172-32-94-150.us-east-2.compute.internal:42159 | Dead | 0 | 0.0 B / 6.2 GB | 0.0 B | 8 | 0 | 0 | 0 | 0 | 0 ms (0 ms) | 0.0 B | 0.0 B | 0.0 B | stdout stderr | | | | | | | | | | | | |
| Showing 1 to 10 of 10 entries Previous 1 Next | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Spark SQLのクエリ実行プランの確認

explain APIよりも正確な、実際に実行されたプランが確認できる

The screenshot displays the Apache Spark SQL interface. On the left, the 'Details for Query 10' section shows a flow diagram: 'WholeStageCodegen' (4.3 m) leads to 'Scan csv' (9,834,292 rows), which leads to 'Execute InsertIntoHadoopFsRelationCommand' (164 files). The main area shows 'Completed Queries (10)' and 'Failed Queries (1)'. An orange arrow points to the failed query (ID 2) in the 'Failed Queries (1)' table. Below, the 'Details' section shows the logical, analyzed, optimized, and physical plans for the failed query.

| ID | Description | Submitted | Duration | Job IDs |
|----|---|---------------------|----------|---------|
| 10 | parquet at NativeMethodAccessorImpl.java:0 | 2021/06/09 06:36:25 | 5.0 min | [18] |
| 9 | csv at NativeMethodAccessorImpl.java:0 | 2021/06/09 06:34:32 | 0.3 s | [16] |
| 8 | parquet at NativeMethodAccessorImpl.java:0 | 2021/06/09 06:28:38 | 5.4 min | [15] |
| 7 | csv at NativeMethodAccessorImpl.java:0 | 2021/06/09 06:26:54 | 8 s | [13] |
| 6 | parquet at NativeMethodAccessorImpl.java:0 | 2021/06/09 06:21:41 | 3.3 min | [12] |
| 5 | csv at NativeMethodAccessorImpl.java:0 | 2021/06/09 06:19:54 | 9 s | [10] |
| 4 | showString at NativeMethodAccessorImpl.java:0 | 2021/06/09 05:51:25 | 35 ms | |
| 3 | showString at NativeMethodAccessorImpl.java:0 | 2021/06/09 05:18:42 | 14 s | [7][8] |
| 1 | csv at NativeMethodAccessorImpl.java:0 | 2021/06/09 05:12:25 | 9 s | [2] |
| | showString at NativeMethodAccessorImpl.java:0 | 2021/06/09 05:08:49 | 12 s | [1] |

| ID | Description | Submitted | Duration | Succeeded Job IDs | Failed Job IDs |
|----|--|---------------------|----------|-------------------|----------------|
| 2 | parquet at NativeMethodAccessorImpl.java:0 | 2021/06/09 05:14:17 | 4.2 min | | [4] |

```
== Parsed Logical Plan ==
'InsertIntoHadoopFsRelationCommand s3://chie-us-east-2/tmp/parquet122, false, ['product_category], Parquet, Map(path -> s3://chie-us-east-2/tmp/parquet122), Overwrite, [marketplace, customer_id, review_id, product_id, product_parent, product_title, product_category, star_rating, helpful_votes, total_votes, vine, verified_purchase, review_headline, review_body, review_date]
+- Relation[marketplace#1336,customer_id#1337,review_id#1338,product_id#1339,product_parent#1340,product_title#1341,product_category#1342,star_rating#1343,helpful_votes#1344,total_votes#1345,vine#1346,verified_purchase#1347,review_headline#1348,review_body#1349,review_date#1350] csv

== Analyzed Logical Plan ==
InsertIntoHadoopFsRelationCommand s3://chie-us-east-2/tmp/parquet122, false, [product_category#1342], Parquet, Map(path -> s3://chie-us-east-2/tmp/parquet122), Overwrite, [marketplace, customer_id, review_id, product_id, product_parent, product_title, product_category, star_rating, helpful_votes, total_votes, vine, verified_purchase, review_headline, review_body, review_date]
+- Relation[marketplace#1336,customer_id#1337,review_id#1338,product_id#1339,product_parent#1340,product_title#1341,product_category#1342,star_rating#1343,helpful_votes#1344,total_votes#1345,vine#1346,verified_purchase#1347,review_headline#1348,review_body#1349,review_date#1350] csv

== Optimized Logical Plan ==
InsertIntoHadoopFsRelationCommand s3://chie-us-east-2/tmp/parquet122, false, [product_category#1342], Parquet, Map(path -> s3://chie-us-east-2/tmp/parquet122), Overwrite, [marketplace, customer_id, review_id, product_id, product_parent, product_title, product_category, star_rating, helpful_votes, total_votes, vine, verified_purchase, review_headline, review_body, review_date]
+- Relation[marketplace#1336,customer_id#1337,review_id#1338,product_id#1339,product_parent#1340,product_title#1341,product_category#1342,star_rating#1343,helpful_votes#1344,total_votes#1345,vine#1346,verified_purchase#1347,review_headline#1348,review_body#1349,review_date#1350] csv

== Physical Plan ==
Execute InsertIntoHadoopFsRelationCommand InsertIntoHadoopFsRelationCommand s3://chie-us-east-2/tmp/parquet122, false, [product_category#1342], Parquet, Map(path -> s3://chie-us-east-2/tmp/parquet122), Overwrite, [marketplace, customer_id, review_id, product_id, product_parent, product_title, product_category, star_rating, helpful_votes, total_votes, vine, verified_purchase, review_headline, review_body, review_date]
+- *(1) FileScan csv [marketplace#1336,customer_id#1337,review_id#1338,product_id#1339,product_parent#1340,product_title#1341,product_category#1342,star_rating#1343,helpful_votes#1344,total_votes#1345,vine#1346,verified_purchase#1347,review_headline#1348,review_body#1349,review_date#1350]
  Batched: false, Format: CSV, Location: InMemoryFileIndex[s3a://amazon-reviews-pds/tsv/amazon-reviews_multilingual_DE_v1_00.tsv.gz, s3a://..., PartitionFilters: [], PushedFilters: [], ReadSchema:
  struct<marketplace:string,customer_id:int,review_id:string,product_id:string,product_parent:int,p...
```

ExecutorとDriverのログ



CloudWatchのLog groupsを確認

※Add Jobの画面でContinuous Logging を有効化したときに出力される

CloudWatch > Log groups > /aws-glue/jobs/logs-v2

/aws-glue/jobs/logs-v2

Actions View in Logs Insights Search log group

▼ Log group details

| | | | |
|--------------|----------------|----------------------|--|
| Retention | Creation time | Stored bytes | ARN |
| Never expire | 1 month ago | 7.95 MB | arn:aws:logs:us-east-2:090132730578:log-group:/aws-glue/jobs/logs-v2:* |
| KMS key ID | Metric filters | Subscription filters | Contributor Insights rules |
| - | 0 | 0 | - |

Log streams Metric filters Subscription filters Contributor Insights Tags

Log streams (100+)

By default, we only load the most recent log streams. [Load more](#).

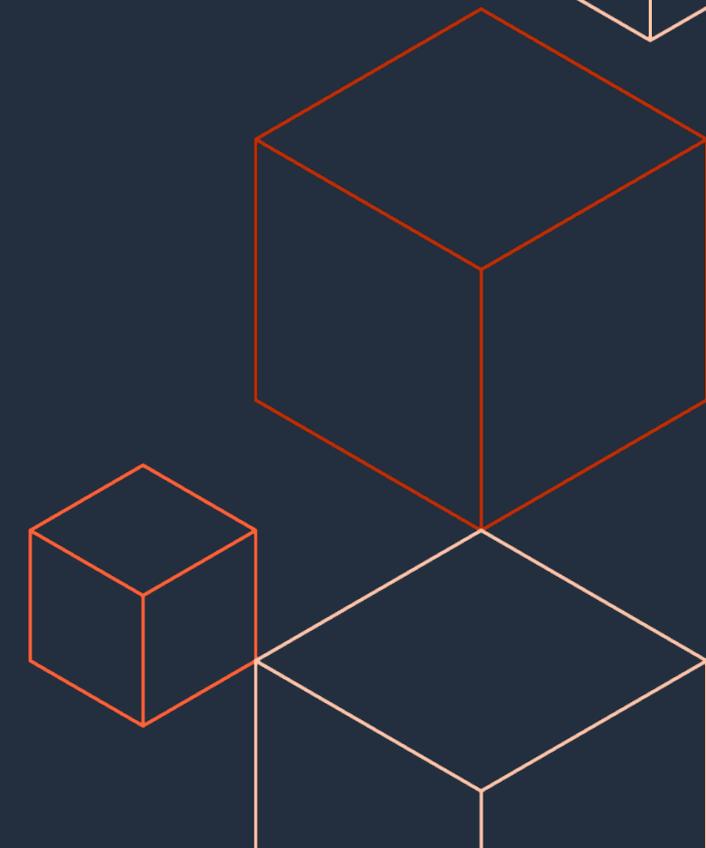
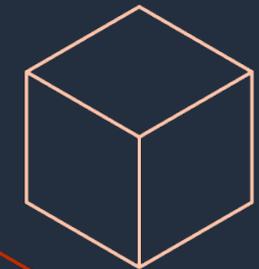
jr_f96500b6ef7e3adb9e4997e7cbde591fedb1a33e50c2e93063620d7177810fb7 11 matches

| Log stream | Last event time |
|--|---------------------------------|
| jr_f96500b6ef7e3adb9e4997e7cbde591fedb1a33e50c2e93063620d7177810fb7-progress-bar | 2021-06-17 03:20:06 (UTC+09:00) |
| jr_f96500b6ef7e3adb9e4997e7cbde591fedb1a33e50c2e93063620d7177810fb7-7 | 2021-06-17 03:20:04 (UTC+09:00) |
| jr_f96500b6ef7e3adb9e4997e7cbde591fedb1a33e50c2e93063620d7177810fb7-9 | 2021-06-17 03:19:56 (UTC+09:00) |
| jr_f96500b6ef7e3adb9e4997e7cbde591fedb1a33e50c2e93063620d7177810fb7-5 | 2021-06-17 03:19:56 (UTC+09:00) |
| jr_f96500b6ef7e3adb9e4997e7cbde591fedb1a33e50c2e93063620d7177810fb7-8 | 2021-06-17 03:19:56 (UTC+09:00) |
| jr_f96500b6ef7e3adb9e4997e7cbde591fedb1a33e50c2e93063620d7177810fb7-4 | 2021-06-17 03:19:55 (UTC+09:00) |
| jr_f96500b6ef7e3adb9e4997e7cbde591fedb1a33e50c2e93063620d7177810fb7-2 | 2021-06-17 03:19:54 (UTC+09:00) |
| jr_f96500b6ef7e3adb9e4997e7cbde591fedb1a33e50c2e93063620d7177810fb7-3 | 2021-06-17 03:19:54 (UTC+09:00) |
| jr_f96500b6ef7e3adb9e4997e7cbde591fedb1a33e50c2e93063620d7177810fb7-6 | 2021-06-17 03:19:54 (UTC+09:00) |
| jr_f96500b6ef7e3adb9e4997e7cbde591fedb1a33e50c2e93063620d7177810fb7-1 | 2021-06-17 03:19:54 (UTC+09:00) |
| jr_f96500b6ef7e3adb9e4997e7cbde591fedb1a33e50c2e93063620d7177810fb7-driver | 2021-06-17 03:19:46 (UTC+09:00) |

Executor ログ

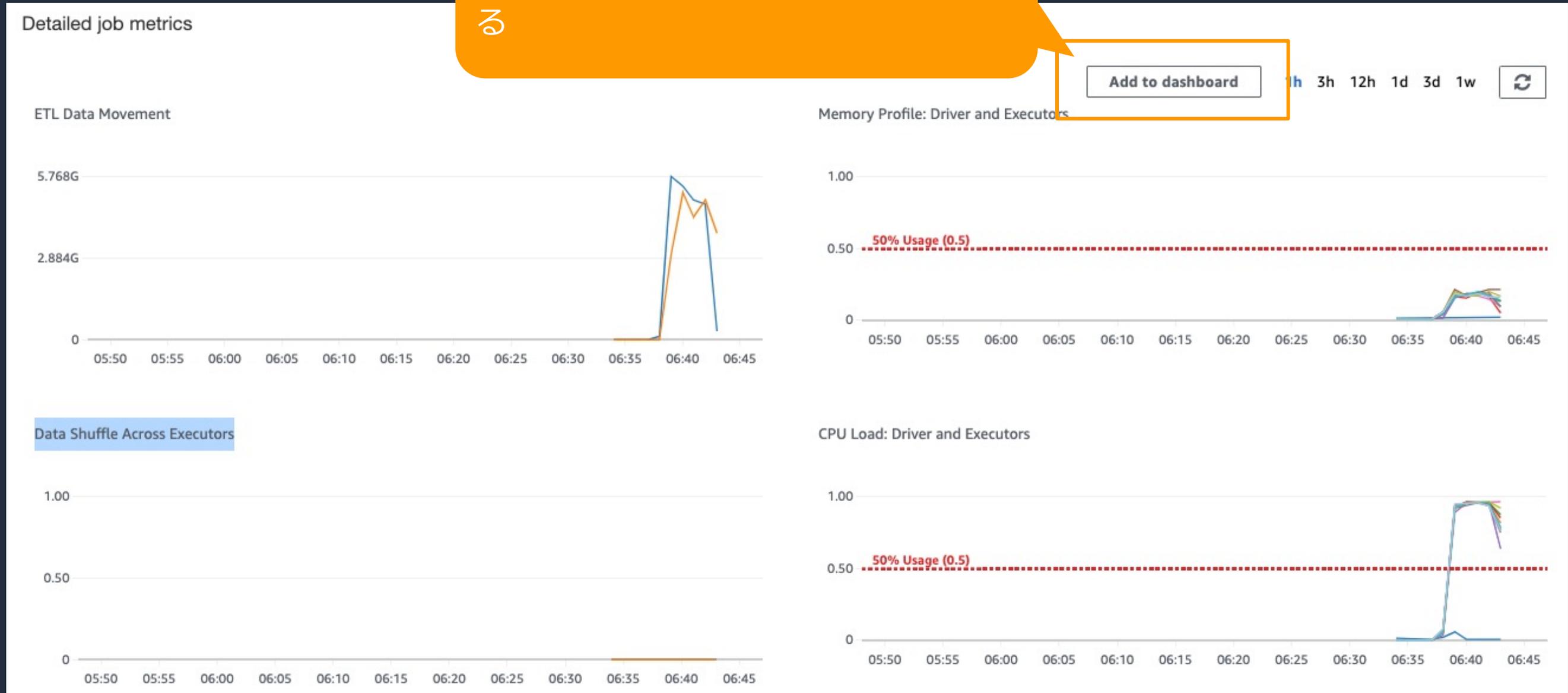
Driver ログ

AWS Glue ジョブメトリクス

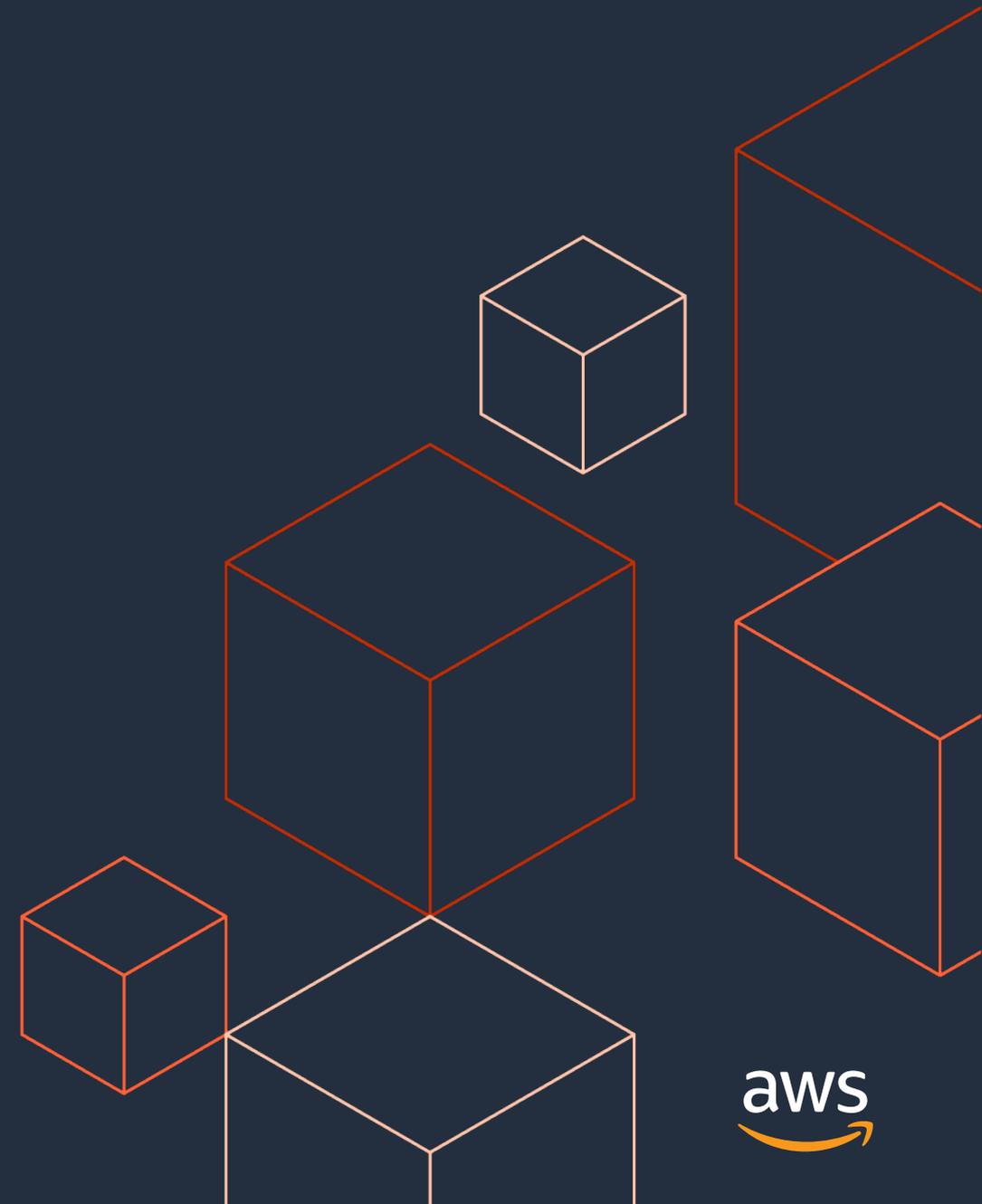


ExecutorやDriverのリソース使用状況を確認

CloudWatchのDashboardを作成し、他のメトリクスも追加することもできる



SparkのAPI



処理途中でのデータの傾向をコマンドで確認する

以下のコマンドを利用して処理途中のデータの傾向を確認して、チューニング戦略に役立てる

※アクションを伴う処理(色文字)は多用すると処理が遅くなるので注意

- `count()`
- `printSchema()`
- `show()`
- `describe([cols*]).show()`
- `explain()`

処理途中でのデータの傾向をコマンドで確認する

df.count()

- レコード数を確認する。
- `df.groupby('col_name').count()`でskewの確認をしたりする。

```
In [4]: df.count()
```

```
9834292
```

```
In [9]: df.groupby(df.product_category).count().show()
```

```
+-----+-----+
| product_category| count|
+-----+-----+
|                PCI| 95783|
|  Lawn and Garden|  1973|
|          Kitchen|  2365|
| Home Entertainment| 37811|
| Home Improvement|  5873|
|             Home|  7008|
|          Wireless| 36228|
|             Video| 56805|
|Digital_Video_Dow...|1116596|
|             Luggage|   475|
|       Video DVD|2066679|
|             Sports|  9694|
|             null|    1|
```

処理途中でのデータの傾向をコマンドで確認する

- `df.printSchema()`
- DataFrameのスキーマ情報を確認する。

```
In [5]: s3_df.printSchema()
```

```
FloatProgress(value=0.0, bar_style='info', description='Progress:', layout=Layout(height='25px', width='50%'),...
```

```
root
```

```
|-- marketplace: string (nullable = true)
|-- customer_id: integer (nullable = true)
|-- review_id: string (nullable = true)
|-- product_id: string (nullable = true)
|-- product_parent: integer (nullable = true)
|-- product_title: string (nullable = true)
|-- product_category: string (nullable = true)
|-- star_rating: string (nullable = true)
|-- helpful_votes: integer (nullable = true)
|-- total_votes: integer (nullable = true)
|-- vine: string (nullable = true)
|-- verified_purchase: string (nullable = true)
|-- review_headline: string (nullable = true)
|-- review_body: string (nullable = true)
|-- review_date: timestamp (nullable = true)
|-- review_date2: date (nullable = true)
|-- review_month: date (nullable = true)
```


処理途中でのデータの傾向をコマンドで確認する

```
df.describe([cols*]).show()
```

- 列ごとの統計値を確認できる。

```
In [18]: df.describe(['customer_id', 'star_rating']).show()
```

| summary | customer_id | star_rating |
|---------|----------------------|--------------------|
| count | 9834292 | 9834291 |
| mean | 3.0019869279621553E7 | 4.311846476782109 |
| stddev | 1.4844815618441587E7 | 1.1425227600410812 |
| min | 10001 | 1 |
| max | 53096591 | 5 |

処理途中でのデータの傾向をコマンドで確認する

df.explain()

- 実行計画を確認できる。

```
In [6]: df.explain()
```

```
== Physical Plan ==
```

```
*(1) FileScan csv [marketplace#20,customer_id#21,review_id#22,product_id#23,product_parent#24,product_title#25,product_category#26,star_rating#27,helpful_votes#28,total_votes#29,vine#30,verified_purchase#31,review_headline#32,review_body#33,review_date#34] Batched: false, Format: CSV, Location: InMemoryFileIndex[s3a://amazon-reviews-pds/tsv/amazon_reviews_multilingual_DE_v1_00.tsv.gz, s3a://..., PartitionFilters: [], PushedFilters: [], ReadSchema: struct<marketplace:string,customer_id:int,review_id:string,product_id:string,product_parent:int,p...
```

```
In [15]: df.explain(extended=True)
```

```
== Parsed Logical Plan ==
```

```
Relation[marketplace#20,customer_id#21,review_id#22,product_id#23,product_parent#24,product_title#25,product_category#26,star_rating#27,helpful_votes#28,total_votes#29,vine#30,verified_purchase#31,review_headline#32,review_body#33,review_date#34] csv
```

```
== Analyzed Logical Plan ==
```

```
marketplace: string, customer_id: int, review_id: string, product_id: string, product_parent: int, product_title: string, product_category: string, star_rating: int, helpful_votes: int, total_votes: int, vine: string, verified_purchase: string, review_headline: string, review_body: string, review_date: timestamp
```

```
Relation[marketplace#20,customer_id#21,review_id#22,product_id#23,product_parent#24,product_title#25,product_category#26,star_rating#27,helpful_votes#28,total_votes#29,vine#30,verified_purchase#31,review_headline#32,review_body#33,review_date#34] csv
```

```
== Optimized Logical Plan ==
```

```
Relation[marketplace#20,customer_id#21,review_id#22,product_id#23,product_parent#24,product_title#25,product_category#26,star_rating#27,helpful_votes#28,total_votes#29,vine#30,verified_purchase#31,review_headline#32,review_body#33,review_date#34] csv
```

```
== Physical Plan ==
```

```
*(1) FileScan csv [marketplace#20,customer_id#21,review_id#22,product_id#23,product_parent#24,product_title#25,product_category#26,star_rating#27,helpful_votes#28,total_votes#29,vine#30,verified_purchase#31,review_headline#32,review_body#33,review_date#34] Batched: false, Format: CSV, Location: InMemoryFileIndex[s3a://amazon-reviews-pds/tsv/amazon_reviews_multilingual_DE_v1_00.tsv.gz, s3a://..., PartitionFilters: [], PushedFilters: [], ReadSchema: struct<marketplace:string,customer_id:int,review_id:string,product_id:string,product_parent:int,p...
```

処理途中でのデータの傾向をコマンドで確認する

df.agg(approx_count_distinct(df.col))

- パーティションキーで利用する列などでカーディナリティを確認する。
- HyperLogLogが利用されるので速い

```
In [21]: df.agg(approx_count_distinct(df.product_category)).show()
```

```
+-----+
|approx_count_distinct(product_category)|
+-----+
|                                     38|
+-----+
```

まとめ

- AWS Glue ETLは Apache Spark というOSSの並列分散処理フレームワークのフルマネージドサービス
- サービスの特性を理解してチューニングを行うことが大切
- ジョブ実行時に収集されるメトリクスを利用してチューニングを行う

『**AWS Glue ETL パフォーマンス・チューニング② チューニングパターン編**』
に続きます。

本資料に関するお問い合わせ・ご感想

- 技術的な内容に関しましては、有料のAWSサポート窓口へお問い合わせください
 - <https://aws.amazon.com/jp/premiumsupport/>
- 料金面でのお問い合わせに関しましては、カスタマーサポート窓口へお問い合わせください（マネジメントコンソールへのログインが必要です）
 - <https://console.aws.amazon.com/support/home#/case/create?issueType=customer-service>
- 具体的な案件に対する構成相談は、後述する個別技術相談会をご活用ください



ご感想はTwitterへ！ハッシュタグは以下をご利用ください
#awsblackbelt

AWSの日本語資料の場所「AWS 資料」で検索



The screenshot shows the AWS Japanese website header with the logo and navigation links. The main heading is 'AWS クラウドサービス活用資料集トップ'. Below it is a paragraph of introductory text. At the bottom, there are four buttons: 'AWS Webinar お申込', 'AWS 初心者向け', 'サービス別資料', and 'ハンズオン資料'.

aws お問い合わせ サポート 日本語 アカウント 今すぐ無料サインアップ

製品 ソリューション 料金 ドキュメント 学ぶ パートナーネットワーク AWS Marketplace イベント さらに詳しく見る

AWS クラウドサービス活用資料集トップ

アマゾン ウェブ サービス (AWS) は安全なクラウドサービスプラットフォームで、ビジネスのスケールと成長をサポートする処理能力、データベースストレージ、およびその他多種多様な機能を提供します。お客様は必要なサービスを選択し、必要な分だけご利用いただけます。それらを活用するために役立つ日本語資料、動画コンテンツを多数ご提供しております。(本サイトは主に、AWS Webinar で使用した資料およびオンデマンドセミナー情報を掲載しています。)

[AWS Webinar お申込](#) [AWS 初心者向け](#) [サービス別資料](#) [ハンズオン資料](#)

<https://amzn.to/JPArchive>

AWSのハンズオン資料の場所「AWS ハンズオン」で検索



The screenshot shows the AWS website's 'AWS Hands-on' page. At the top, there is a navigation bar with the AWS logo, a search bar, and links for 'お問い合わせ', 'サポート', '日本語', and 'アカウント'. A prominent orange button says '今すぐ無料サインアップ'. Below the navigation, a main heading reads 'AWS ハンズオン資料'. The content area is split into two columns. The left column contains the text: 'AWS をステップバイステップでお試しいただくのに役立つ動画および資料を掲載しています。' followed by 'その他の資料は以下をご覧ください。' and two links: '初心者向けの資料' and 'サービス別の資料'. The right column contains two links: 'AWS オンラインセミナースケジュール' and 'AWS クラウドサービス活用資料集トップ'. Below this main content area, there is a white box with the heading 'AWS 初心者向けハンズオン' and a paragraph: 'AWS 初心者向けに「AWS Hands-on for Beginners」と題し、初めて AWS を利用する方や、初めて対象のサービスに触る方向けに、操作手順の解説動画を見ながら自分のペースで進められるハンズオンをテーマごとにご用意しています。'

<https://aws.amazon.com/jp/aws-jp-introduction/aws-jp-webinar-hands-on/>

AWS Well-Architected個別技術相談会

- 毎週「W-A個別技術相談会」を実施中
 - AWSのソリューションアーキテクト（SA）に
対策などを相談することも可能
- 申込みはイベント告知サイトから
 - <https://aws.amazon.com/jp/about-aws/events/>

AWS イベント

で[検索]

ご視聴ありがとうございました