

ApacheSparkThroughEmail1

July 10, 2024

1 Apache Spark Through Email - Part 1 Read, cache, analyze

1.1 Read email from file in Apache Parquet format

```
[1]: val homeDir = scala.sys.props("user.home")
      val records = spark.read.parquet(s"$homeDir/datasets/enron/enron-small.parquet")
```

```
homeDir = /home/medale
records = [uuid: string, from: string ... 8 more fields]
```

```
[1]: [uuid: string, from: string ... 8 more fields]
```

1.2 Cache dataset for repeated exploration

Lazy evaluation of transformations vs. action

```
[2]: // read was a transformation - nothing executed yet
      records.cache //a transformation - nothing gets executed yet
```

```
[2]: [uuid: string, from: string ... 8 more fields]
```

```
[3]: records.count //an action - actually read data and perform "cache"
      ↪ transformation
```

```
[3]: 191926
```

Records now cached in memory on Spark standalone executor(s) - see <http://localhost:8080/> - Running Application - Application Detail UI

```
[4]: // cached in memory - fast
      records.count
      records.printSchema
```

```
root
 |-- uuid: string (nullable = true)
 |-- from: string (nullable = true)
 |-- to: array (nullable = true)
 |    |-- element: string (containsNull = true)
```

```

|-- cc: array (nullable = true)
|   |-- element: string (containsNull = true)
|-- bcc: array (nullable = true)
|   |-- element: string (containsNull = true)
|-- dateUtcEpoch: long (nullable = true)
|-- subject: string (nullable = true)
|-- mailFields: map (nullable = true)
|   |-- key: string
|   |-- value: string (valueContainsNull = true)
|-- body: string (nullable = true)
|-- attachments: array (nullable = true)
|   |-- element: struct (containsNull = true)
|   |   |-- fileName: string (nullable = true)
|   |   |-- size: integer (nullable = true)
|   |   |-- mimeType: string (nullable = true)
|   |   |-- data: binary (nullable = true)

```

1.3 Who are the Top 10 email message senders?

```

[5]: //https://spark.apache.org/docs/latest/api/scala/index.html#org.apache.spark.
      ↪sql.functions$
import org.apache.spark.sql.functions.desc

//lazy transformations
val fromsWithCountsDesc = records.select("from").
  groupBy("from").
  count.
  orderBy(desc("count")).
  limit(10)

//action
fromsWithCountsDesc.show(truncate = false)

```

```

+-----+-----+
|from                    |count|
+-----+-----+
|vince.kaminski@enron.com|14340|
|jeff.dasovich@enron.com |10888|
|chris.germany@enron.com  |8688 |
|steven.kean@enron.com    |6722 |
|sally.beck@enron.com     |4253 |
|john.arnold@enron.com    |3505 |
|david.delainey@enron.com |2991 |
|enron.announcements@enron.com|2803 |
|pete.davis@enron.com     |2753 |
|phillip.allen@enron.com  |2145 |
+-----+-----+

```

```
fromsWithCountsDesc = [from: string, count: bigint]
```

```
[5]: [from: string, count: bigint]
```

```
[6]: records.unpersist
```

```
[6]: [uuid: string, from: string ... 8 more fields]
```

1.4 Kill the Spark application that was created for running this notebook

```
[7]: spark.close
```

```
[ ]:
```