APACHE Spark

LOAD BIG DATA EFFICIENTLY

PART 1: HOW DIFFERENT DATA FORMATS BEHAVE IN ACTION

DATA WITH NIKK THE GREEK

APACHE Spark™

{json}

- ***Evalute file formats based on:***
  - ***Size***
  - ***Saving time***
  - ***Load time***

CSV

Parquet

AVRO™

DATA WITH NIKK THE GREEK

In VSCode

Format	Size	Write time VS Code	Load time VS Code
JSON	1208 MB	6.6 s	11.7 s
CSV	593 MB	6.2 s	5.3 s
CSV with infer schema	593 MB	6.2 s	23.5 s
PARQUET	81.5 MB	5.4 s	1.5 s
AVRO	69.2 MB	2.5 s	2.3 s

In Spark UI

Format	Load time Spark UI	of which meta data	of which actual load
JSON	11 s	4 s	7 s
CSV	5 s	45 ms	5 s
CSV with infer schema	23 s	20 ms + 11 s	12 s
PARQUET	1.1 s	98 ms	1 s
AVRO	2 s	0 s	2 s

Observations

- Avro and Parquet seem to be highly compressed and significant smaller than CSV and JSON
- Despite compression writes are up to 50 % faster, reads also significantly faster than CSV and JSON. Avro writes faster than parquet thow
- Parquet and Avro seem to provide the schema correctly. CSV and Json don‘t. Json seem not to understand timestamps.
- Parquet seem to contain meta data due to scanning activity in SQL and also a read job ahead
- CSV and JSON have a preliminary job interfering the schema
- Avro provides schema correctly without any preliminary job