

Apache Spark

Day 2

EcZachly Inc





EcZachly Inc

Spark Server vs Spark Notebooks

- Spark Server (how Airbnb does it)
 - Every run is fresh, things get uncached automatically
 - Nice for testing
- Notebook (how Netflix does it)
 - Make sure to call unpersist()



EcZachly Inc

Databricks considerations

- Considerations
 - Databricks should be connected with Github
 - PR review process for EVERY change
 - CI/CD check



EcZachly Inc

Caching and Temporary Views

- Temporary views
 - ALWAYS GET RECOMPUTED UNLESS CACHED
- Caching
 - Storage Levels
 - MEMORY_ONLY
 - DISK_ONLY
 - MEMORY_AND_DISK (the default)
 - Caching really only is good if it fits into memory.
 - Otherwise there's probably a staging table in your pipeline you're missing!
 - In notebooks,
 - Call unpersist when you're done otherwise the cached data will just hang out!



EcZachly Inc

Caching vs Broadcast

- Caching
 - Stores pre-computed values for re-use
 - Stays partitioned
- Broadcast Join
 - Small data that gets cached and shipped in entirety to each executor



EcZachly Inc

Broadcast JOIN optimization

- Broadcast JOINS prevent Shuffle
 - GOOD!
- The threshold is set
 - `spark.sql.autoBroadcastJoinThreshold`
- You can explicitly wrap a dataset with `broadcast(df)` too!

UDFs



EcZachly Inc

PySpark quirks vs Scala

Apache Arrow optimizations in recent versions of Spark have helped PySpark UDFs become more inline with Scala Spark UDFs

Dataset API allows you to not even need UDFs, you can use pure Scala functions instead!



EcZachly Inc

DataFrame vs Dataset vs SparkSQL

Dataset is Scala only!

DataFrame vs SparkSQL:

- DataFrame is more suited for pipelines that are more hardened and less likely to experience change
- SparkSQL is better for pipelines that are used in collaboration with data scientists
- Dataset is best for pipelines that require unit and integration tests.



EcZachly Inc

Parquet

- An amazing file format
 - [Run-length encoding](#) allows for powerful compression
- Don't use global `.sort()`
 - Painful, slow, annoying
- Use `.sortWithinPartitions`
 - Parallelizable, gets you good distribution

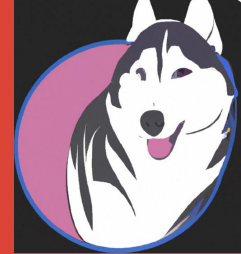
Spark Tuning



EcZachly Inc

- Executor Memory
 - Don't just set to 16 GBs and call it a day, wastes a lot
- Driver Memory
 - Only needs to be bumped up if:
 - You're calling `df.collect()`
 - Have a very complex job!

Spark Tuning



EcZachly Inc

- Shuffle Partitions
 - Default is 200
 - Aim for ~100 MBs per partition to get the right sized output datasets!
- AQE (adaptive query execution)
 - Helps with skewed datasets, wasteful if the dataset isn't skewed



EcZachly Inc

Lab structure

- Going over an example on caching
- Doing a job with each:
 - DataFrame vs Dataset vs SparkSQL
- Showing how bucketing works in Iceberg