ippon POSITIVE TECHNOLOGY

Build next generation Big Data applications with Delta Lake

AGENDA

PART 01
INTRO
WHAT and WHY
Delta Lake

PART 02
DEEP DIVE
&
FEATURES

PART 03
DEMO
CONCLUSION
Q&A

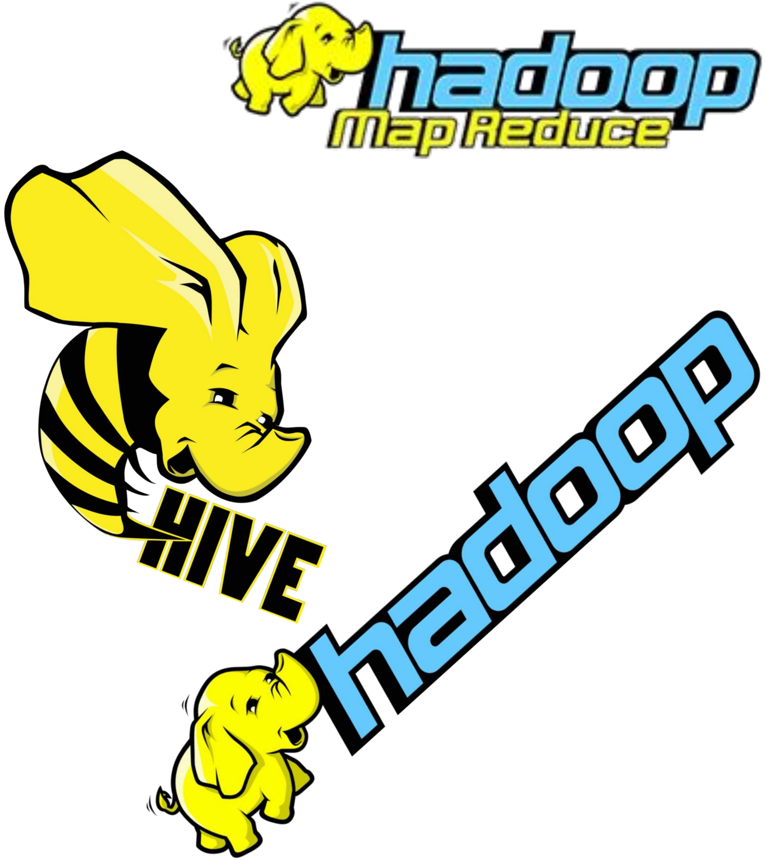
INTRODUCTION
Théo Lebrun

- [Falydoor]
- [@Falydoor]
- [https://theolebrun.com/]
- Data Engineer and Technical Manager at [Ippon Technologies]

Let’s talk about the past

- **Hadoop ecosystem**
  - HDFS
  - YARN
  - MapReduce

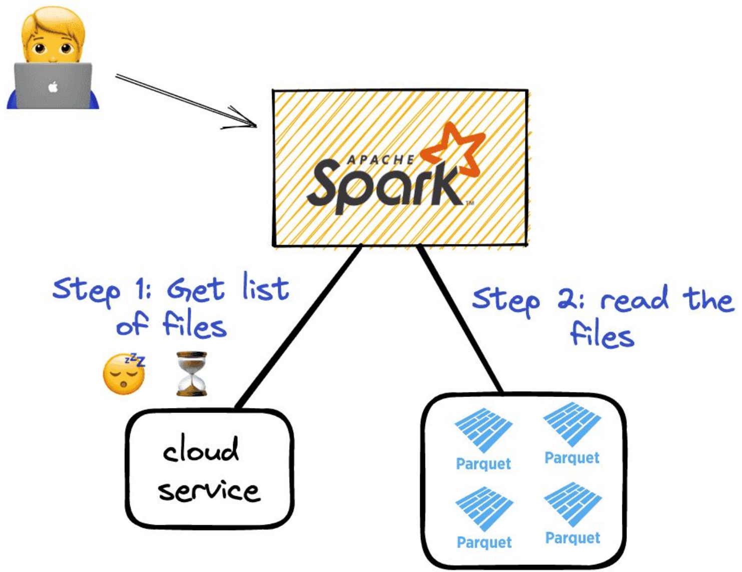
- **Can’t look at data directly**
- **Require binaries (hadoop or hdfs)**
- **17 years old**



What about Data Lake?

- **Structured or unstructured data (CSV, JSON, PDF, etc)**
- **Cloud or on-prem**
- **Will quickly become a “Data Swamp”**
- **Parquet lake is not ideal**
  - No ACID transactions
  - Parquet files are immutable
  - Schema updates requires a rewrite

Data Lake File Listing operations are slow

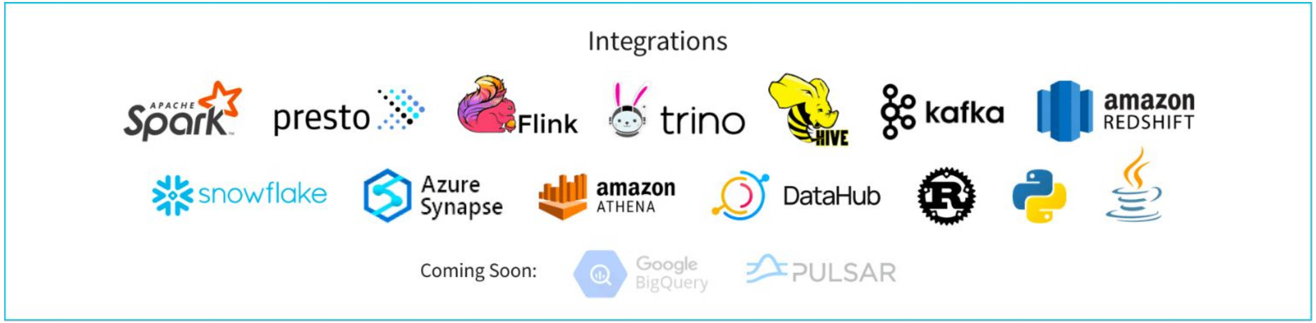


Introducing Delta Lake!

- Transaction log with metadata
- File skipping
- Partitioned data
- Z-order
- Interoperability
- Deletion vectors

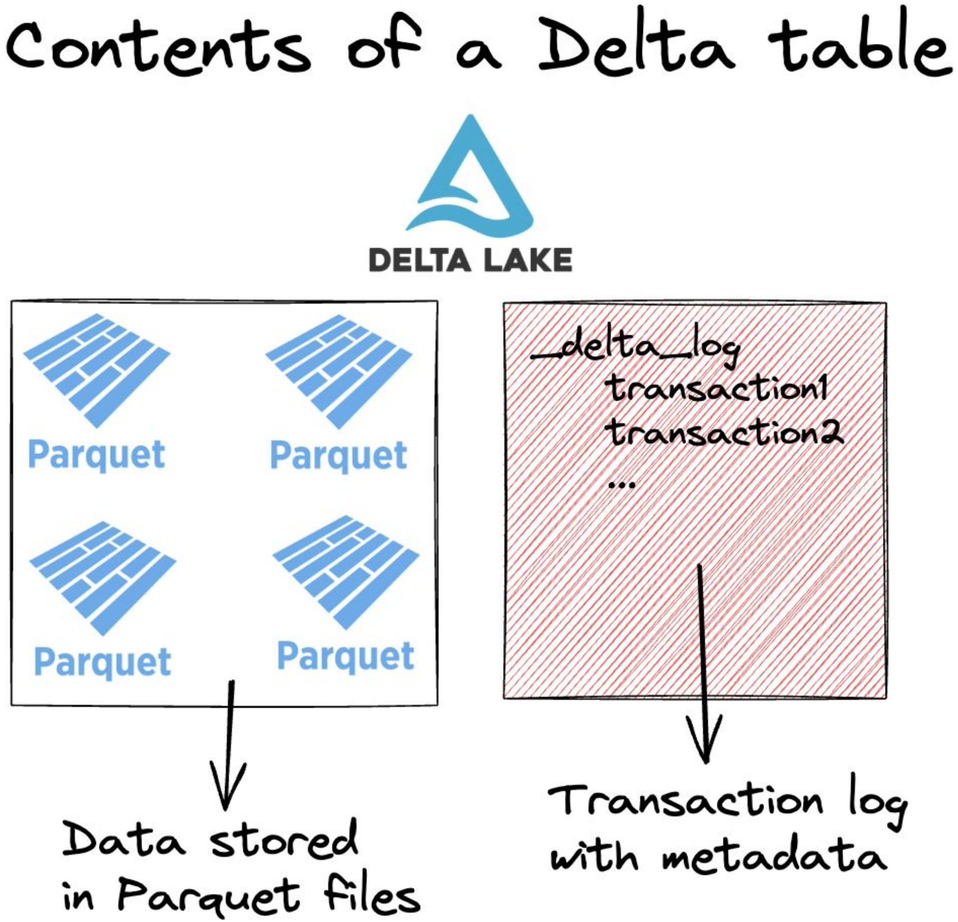
DELTA LAKE

THE APACHE SOFTWARE FOUNDATION



Delta Table

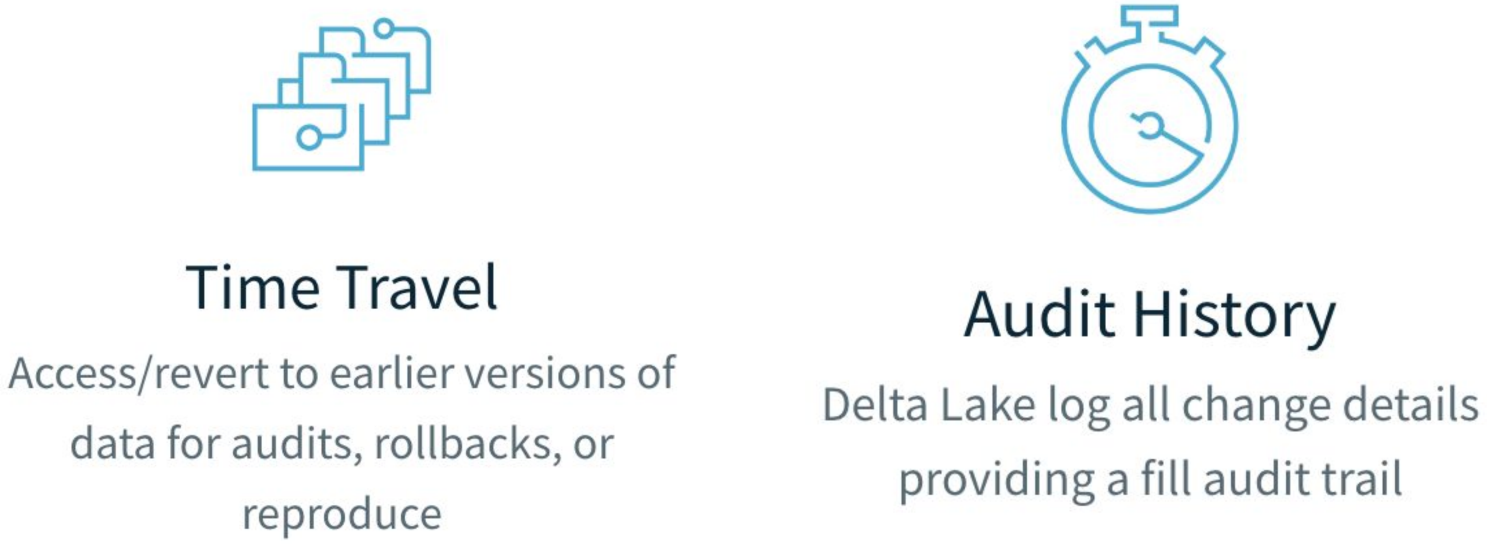
- **ACID Transactions**
- **Unified Batch/Streaming**
- **DML Operations**
    - Create, update and delete
    - SQL, Java, Scala, and Python
- **Scalable Metadata**


Contents of a Delta table
DELTA LAKE
Parquet
Parquet
Parquet
Parquet
_delta_log
transaction1
transaction2
...
Data stored in Parquet files
Transaction log with metadata

Schema Evolution

- **Logical column name**
- **Quick column drop**
- **Schema enforcement**
- **Check constraints**

Time Travel and Audit



Time Travel

Access/revert to earlier versions of data for audits, rollbacks, or reproduce

Audit History

Delta Lake log all change details providing a fill audit trail

Optimize and Vacuum

- Small file problem (ideal size is ~500MB per file)
- Compaction or bin-packing for the win
- Vacuum can save you $$$
- 7 days history by default

Demo

- Ingest Data and create a Delta table
- Update the schema
- Perform a merge
- Run utilities like optimize and vacuum
- Read Data with Polars

Delta Lake ecosystem

- Polars - https://docs.pola.rs/py-polars/html/reference/io.html#delta-lake
- Python Delta Lake and Rust Delta Lake - https://delta-io.github.io/delta-rs/
- Delta Sharing - https://delta.io/sharing
- Protocol - https://github.com/delta-io/delta/blob/master/PROTOCOL.md
- Delta Lake website - https://delta.io/

Thank you for your time!

I would love to chat more with you about building a successful Data platform using Delta Lake or any other technology.

You can email me at tlebrun@ipponusa.com or just come talk to me after this presentation.

ippon

Q&A

en.ippon.tech

contact@ipponusa.com —— +1 844-477-6687 —— @ipponUSA