



ETL Framework


FOR BIG DATA - POWERED BY APACHE SPARK

Agenda

- ▶ The Motivation
- ▶ Why another ETL framework ?
- ▶ Working of the ETL framework
- ▶ Not a bunch of API's
- ▶ Vision and Roadmap
- ▶ Where are we now ?
- ▶ Conclusion

The Motivation

- ▶ Technical Designer (Big Data) for a large Government Account in UK.
- ▶ Responsible for creating system to mimic Dimension Modelling on Hadoop. (Data Warehouse on Big Data).
- ▶ Data model had Reference Tables, Dimensions, Facts
- ▶ Data Repository on Hive. Data transformation using Spark.
- ▶ Batch Jobs created using Apache Spark, Hive. Worked with Data Modeller for mapping complex Oracle SQL to SparkSQL.

- 
- ▶ We cannot escape Dimensional Modelling – Data Modeller
 - ▶ Each batch job on an average had 500 Lines of code. 300 Data transformation, rest were Dimensional model based logic- Primary Keys generation, Complex joins, Slowly Changing Dimensions, Foreign Keys and Spark / User Variables.
 - ▶ Around 20 Spark Jobs created to cover the scope of the Project.
 - ▶ Patterns begin to appear in the Jobs. Grouped the functionality and created a Template. Made API available for achieving the functionality.
 - ▶ Templates for Primary Key, Lookups, Joins, SCD 1 and 2 were created in version 0.1
 - ▶ The Spark Jobs were refactored to include the Framework.

Why another ETL Framework ?

- ▶ Market Leaders and Commercial Vendors in Data Integration like Informatica, IBM Datastage (both are ETL) have proprietary engines for Data Transformation. Oracle Data Integrator (ELT) rely on underlying database for data transformation.
- ▶ Adapted or Improved for Push down processing (on Hadoop cluster). Performance often restricted since Push down feature depends on execution engine of Hadoop i.e. MapReduce, Tez, Hive on Spark. But don't support Spark as execution engine. They are slow in adopting latest improvement in Hadoop or sometimes don't support certain products of the Hadoop ecosystem.
- ▶ Talend (an exception) can process with Spark as Execution engine, but features is present in the Commercial Edition.
- ▶ We don't want to compare ETL framework against established ETL Vendors (at earlier stages)

Working of ETL Framework (PK)

- ▶ Developer includes the ETL Framework JAR's in the Eclipse environment.
- ▶ Focus of building Business/Transformation logic in Spark
- ▶ Transform RDD's and create the Dataframe.
- ▶ Dataframe to be stitched with Primary/ Surrogate Keys.

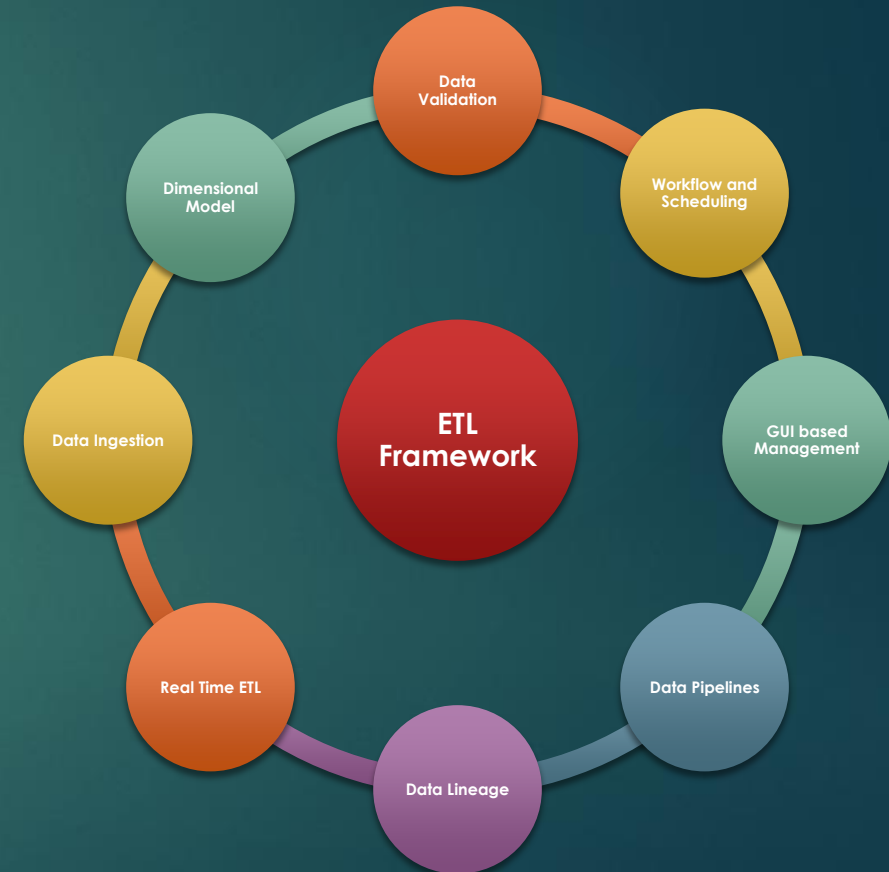
Not Another Bunch of API's

- ▶ The ETL framework frees the developer with nitty-gritty of Dimensional modelling and let them focus on building Business logic.
- ▶ Created using Spark 1.6.1 .Set of JAR's to be imported.
- ▶ Easily extensible. Complex logic made Simple.
- ▶ But is it a ETL Framework ?
- ▶ The API provide Data Transformation and Target functionality, no Extraction logic.
- ▶ To built a complete ETL Framework, we need more features: Low Source System Impact, Highly Performance, Data Validation, Logging, Configurable Built in Error Handling, Customizable, Re-Runnable, Meta Data Driven Code Generation, Unit Testing Capabilities for Development

Vision

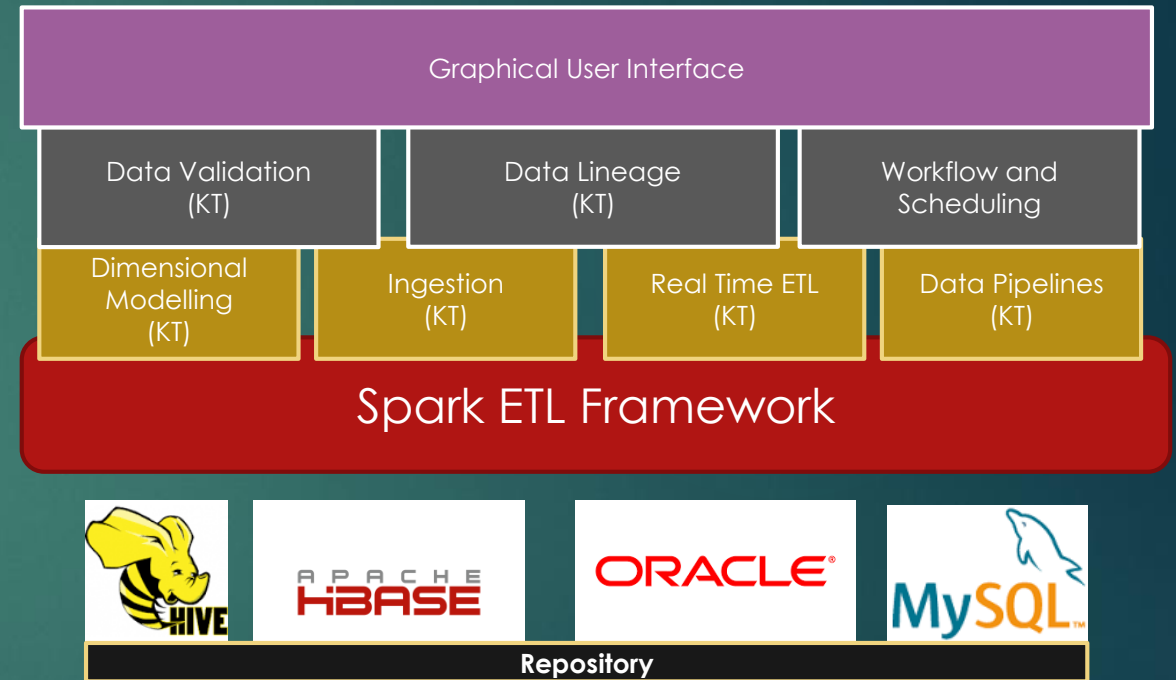
- ▶ Design a unique and complete ETL framework
- ▶ Platform for Developer to Speed up the Development Lifecycle.
- ▶ Customized Templates for Data Ingestion (Extraction) capability with Real Time Ingestion.
- ▶ Customized Template for Dimensional Modelling capabilities plus Real Time Transformation.
- ▶ Rule Based Template for Data Validation, API for Real Time Rule checking.
- ▶ Data Lineage - Source to Data Mart.
- ▶ E2E Scheduling capabilities.
- ▶ Open Source the Framework.

Most of features already available in Apache world



Unified Framework

- ▶ The ETL Repository can be RDBMS or Hadoop Ecosystem.
- ▶ Knowledge API/Templates provides the core functionality.
- ▶ GUI to manage the KPI.



Where are we now ?

- ▶ v0.1 ready for Dimensional Templates
- ▶ v0.1 for Ingestion Templates in progress (Oracle only)
- ▶ Stitching 2 modules using GUI (low priority)
- ▶ February 6th 2017 - Deadline for v0.1 for Ingestion and Transformation

