

The Automation of the Data Lake Ingestion Process from Various Sources

Aleksandar Tunjić

Multicom d.o.o., Zagreb, Croatia

aleksandar.tunjic@multicom.hr



Introduction

- Part of a larger project – primary and secondary education in Croatia

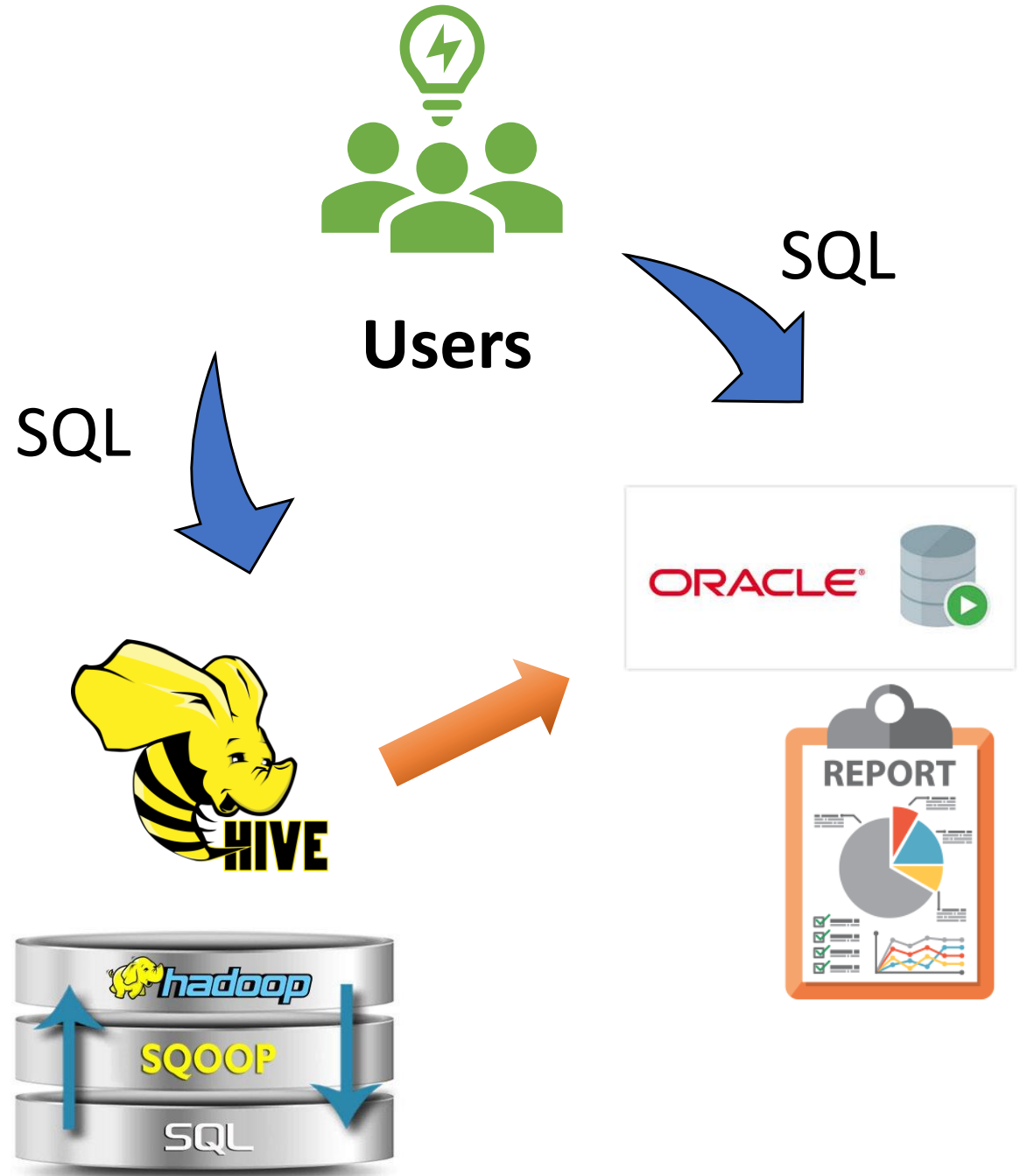


Users



Goal

- Integrate data into Hadoop
- Users can query Hadoop data using SQL
- Export to data warehouse





Hive

- Hive and Impala support standard SQL
- Parquet table format on HDFS
- Can be read by Pig and Map Reduce
- Supports partitions

- Limitations:
- No support for UPDATE/DELETE

- However, no need for UPDATE/DELETE
- INSERT and DROP PARTITION are sufficient

Limitations of the project

Avoid additional costs and dependency on:

- Oracle Big Data Connectors
 - Oracle Data Integrator
 - Oracle Golden Gate
 - Etc...
-
- Therefore, we decided to use Apache Sqoop for data transfers

cloudera[®]



Apache Sqoop



Import an entire table from RDBMS to HDFS:

```
sqoop import \  
--connect jdbc:mysql://mysql.example.com/sqoop \  
--username sqoop \  
--password sqoop \  
--table cities \  
--target-dir /etl/input/cities \  
--where "country = 'USA'" \  
--num-mappers 10
```





Sqoop limitations

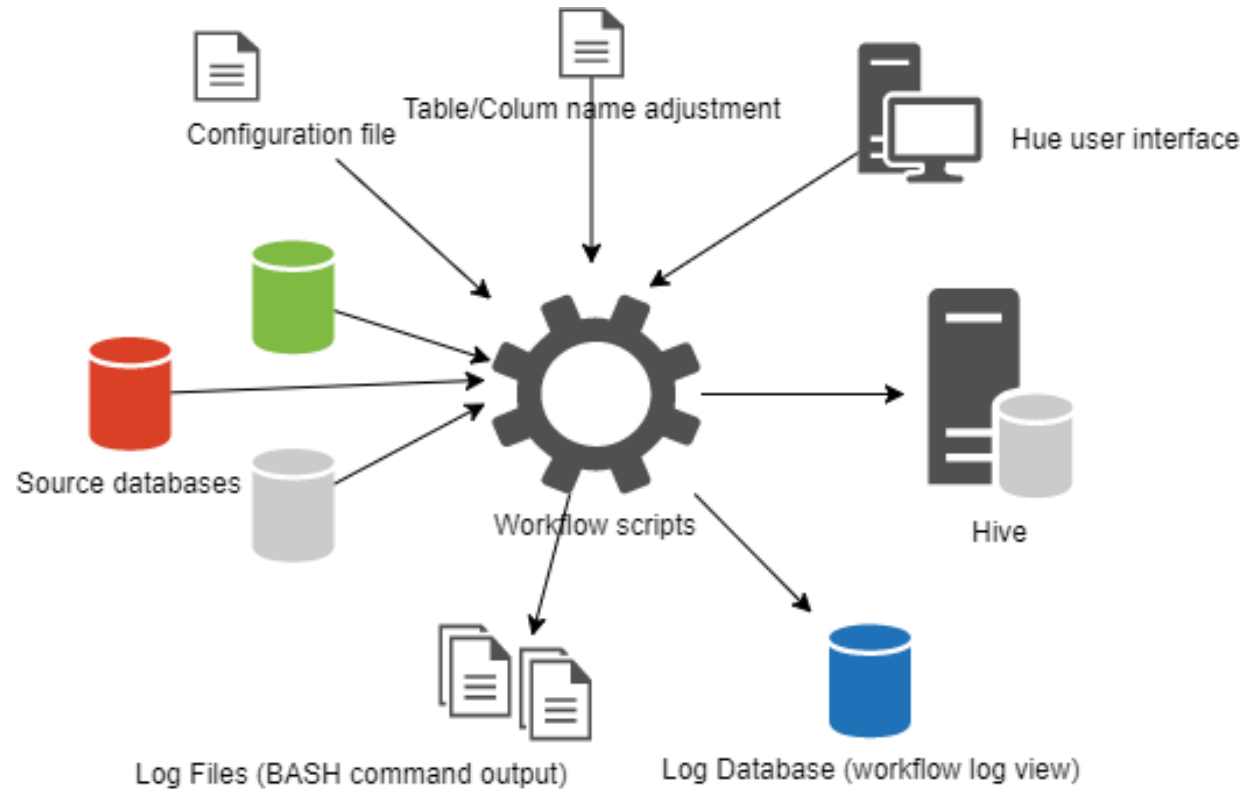
- Keywords in source columns (e.g. „order“)
- Special characters in source columns or table names
- Importing date/datetime/timestamp columns – bug?
- We need a new column – „LOAD_DATE“

Solution: Free-form queries

- More limitations:
 - Import only 1 table at a time
 - import into partitioned Hive tables not supported

Project Requirements

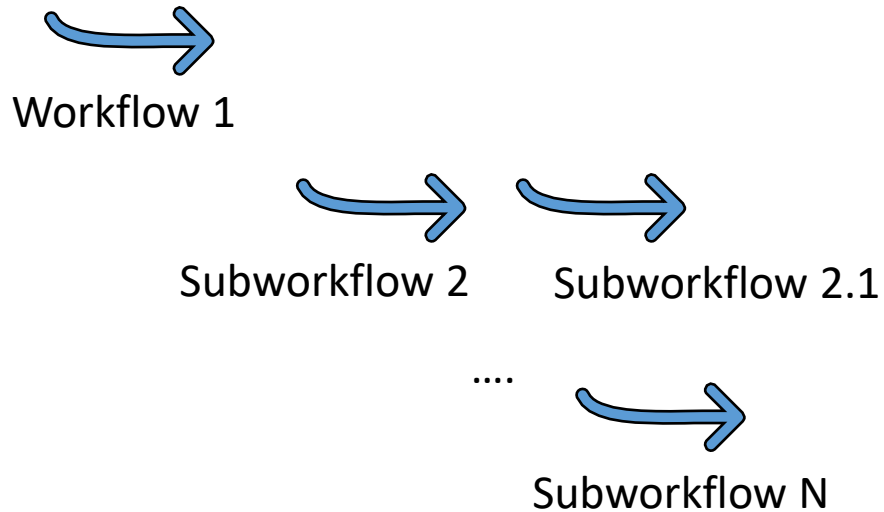
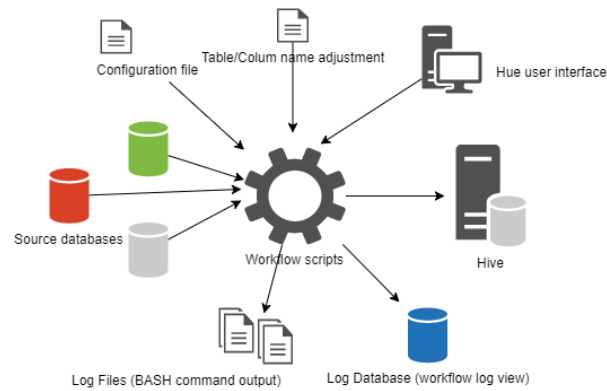
- A periodical import of data from Postgres, MySQL and SQL Server databases into Hive tables.
- Processes must be scheduled so they can be automatically started at a defined time.
- Tables for import selected by user.
- Full copies or copies of incremental changes where possible
- Filter data before importing (in the WHERE clause).
- Table format supports partitioning and is readable by Impala as well.
- Since there are many tables, they need to be created automatically using metadata from RDBMS
- There needs to be a log that contains a status for each process.



- Atomic units of work
- Written as BASH scripts
- Nested workflows

- 3 types:
 - Create Hive table
 - Import data
 - Export data
- Controlled by configuration files
- Write status to a log table

Workflow design



- Workflows can be nested
- Subworkflows can be started sequentially or in parallel
- Workflow states:
 - Success
 - In process
 - Error
- States are stored in an external RDBMS table (log)

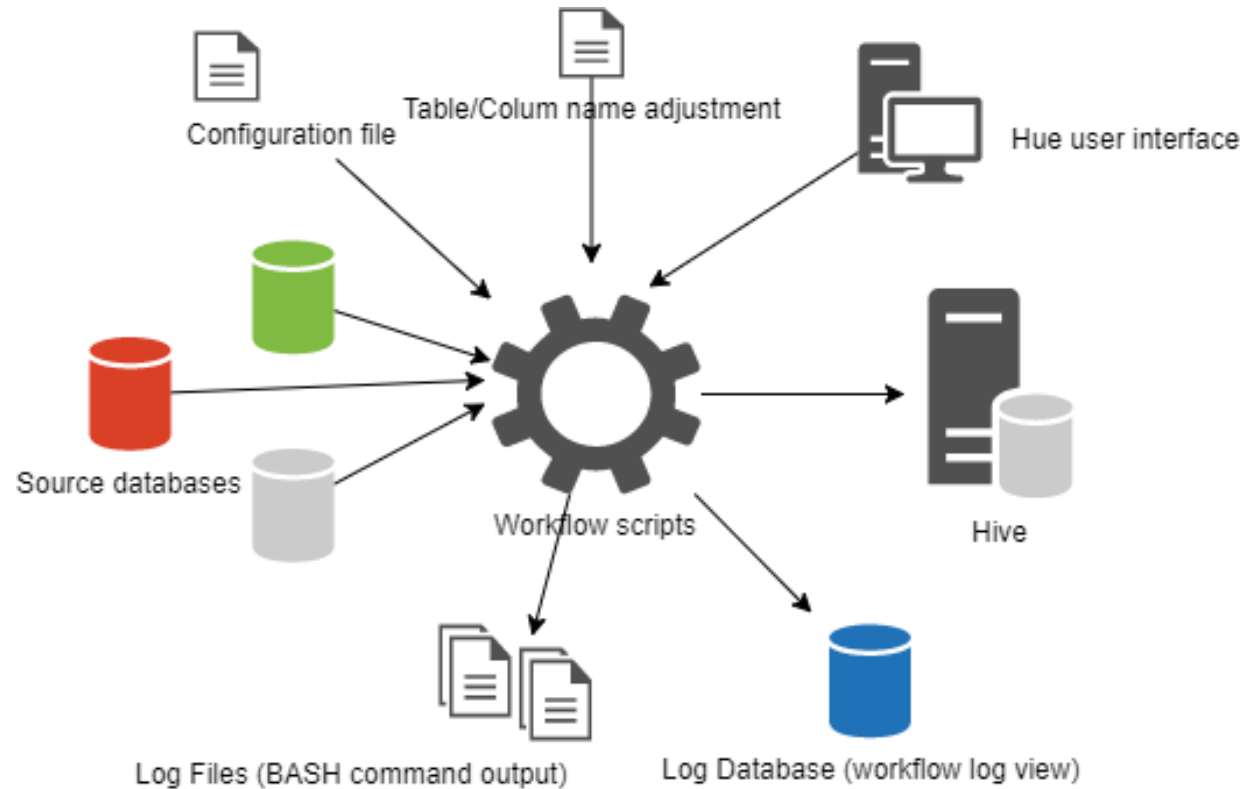
Workflow design - continued

Workflow for creating tables

- Potentially thousands of tables at source (automation!)
- Sqoop create-hive-table tool (limitations!)

We need to:

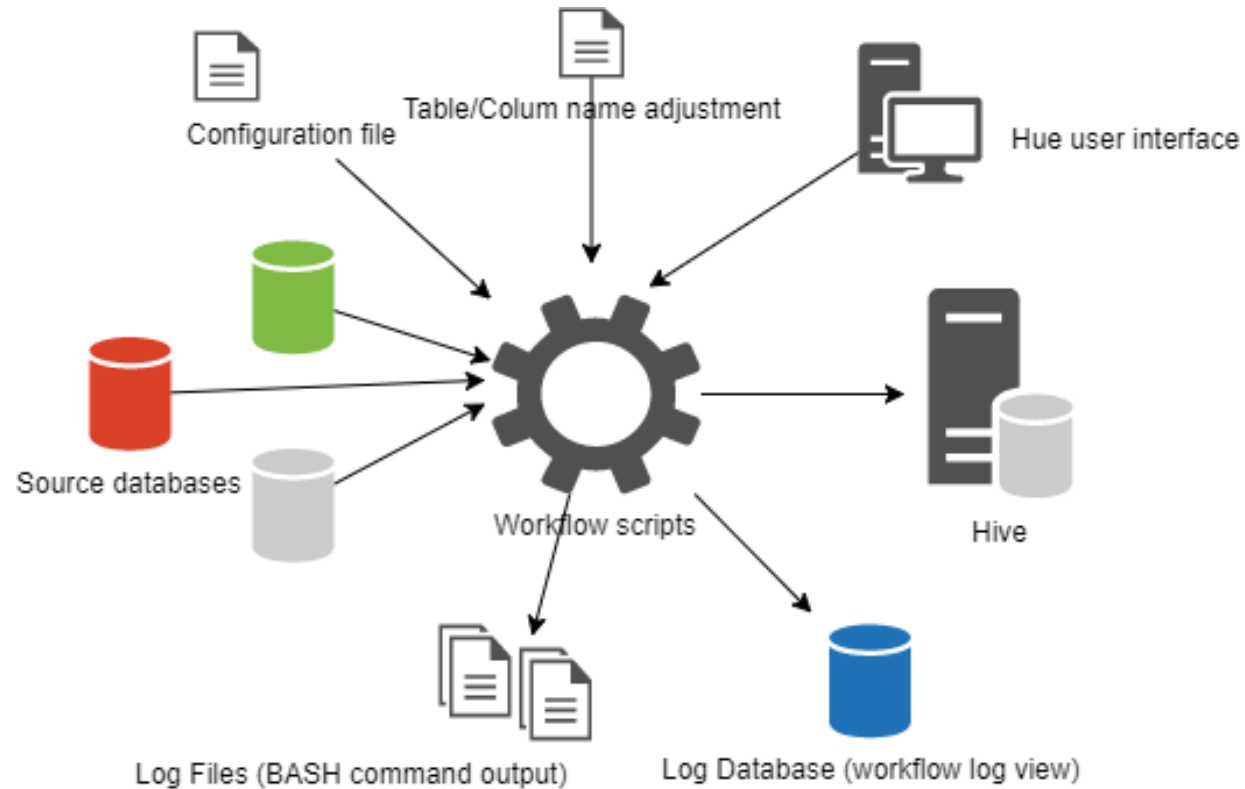
- Append at least 1 new column to Hive table (LOAD_DATE)
- Handle timestamps
- Adjust column/table names for Hive compatibility
- Create partitioned tables in Parquet



Two iterations:

1. Create **temporary** databases and tables with in Hive (sqoop-import tool)
2. Create final databases and tables in Hive
 - Using Hive's Beeline tool (DESCRIBE TABLE, CREATE TABLE statements) by reading metadata from temporary tables
 - Add LOAD_DATE and columns for timestamp reformatting

Workflow for creating tables



- Incremental or full import
- Sqoop-import tool with free-form query

1. Read metadata from Hive with Beeline
2. Construct SQL query (increment?)
3. Import data to HDFS
4. Import into Hive using Beeline (LOAD DATA INPATH)

Workflows for importing data

>_ Shell

The screenshot shows the Hue Shell interface. At the top, there is a text input field containing the path `/user/yarn/postgres/full_import/`. Below this, the interface is divided into two columns: "ARGUMENTS +" and "FILES +". Each column contains a list of items, each with a plus icon on the left, a minus icon on the right, and a file icon on the far right. The "ARGUMENTS" column contains: `DATABASE1_FULL_IMPORT`, `${load_date}`, `10`, `/etc/hive/conf`, `daf1node05-adm.multicom.t`, `10000|` (highlighted with a blue border), `load_date`, and `77.198.243.30`. The "FILES" column contains: `/user/yarn/postgre`, `/user/yarn/postgre`, `/user/yarn/postgre`, `/user/yarn/postgre`, `/user/yarn/postgre`, `/user/yarn/postgre`, `/user/yarn/postgre`, and `/user/yarn/postgre`. Each file entry also has a double-dot icon between the file path and the file icon.

Hue (Oozie) Workflows

- Hue provides user-friendly interface
- Part of Cloudera Hadoop distribution
- „Shell action” – interface for starting BASH scripts

Which workflow to schedule?

eDnevnik Import Workflow 



How often?



Every at :

 Hide

Advanced syntax

Timezone 

From  

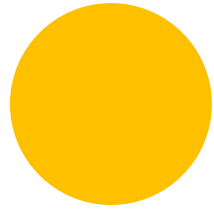
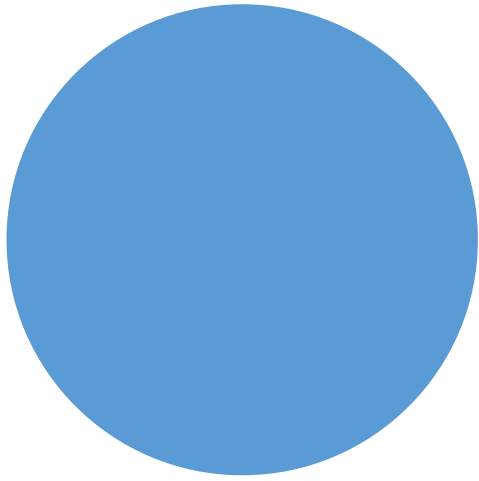
To  

Parameters

Hue (Oozie) Schedules

- In Hue we can schedule the Oozie workflow from previous slide
- User-friendly interface for one-time or repeating schedules
- Forward parameters to workflows (LOAD_DATE in our case...)



Aleksandar Tunjić

Multicom d.o.o., Zagreb, Croatia

aleksandar.tunjic@multicom.hr

Thank you!

Questions?

