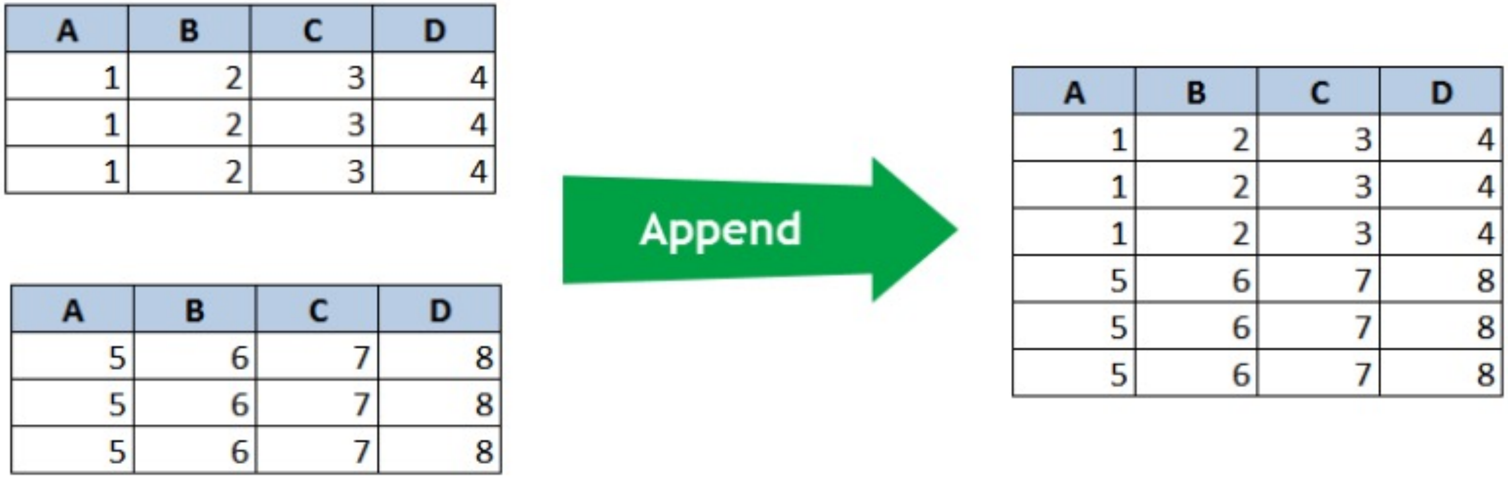
Implementation of SparkR within PDC

Author: Ahmed, Ifraz

Introduction

Background of PDC

- PDC (Private Debt Collection) is a program within the IRS to collect outstanding taxes owed using PCAs (Private Collection Agencies).
- The purpose of the RAAS PDC program is to enable and inform programmatic decision-making through strategic and analytical support.
- Data collection within PDC started in April 2017, and are appended to the Entity History Table (EHIST).
- The amount of data recently outgrew the resources available, creating a need for a big data solution.



A	B	C	D
1	2	3	4
1	2	3	4
1	2	3	4

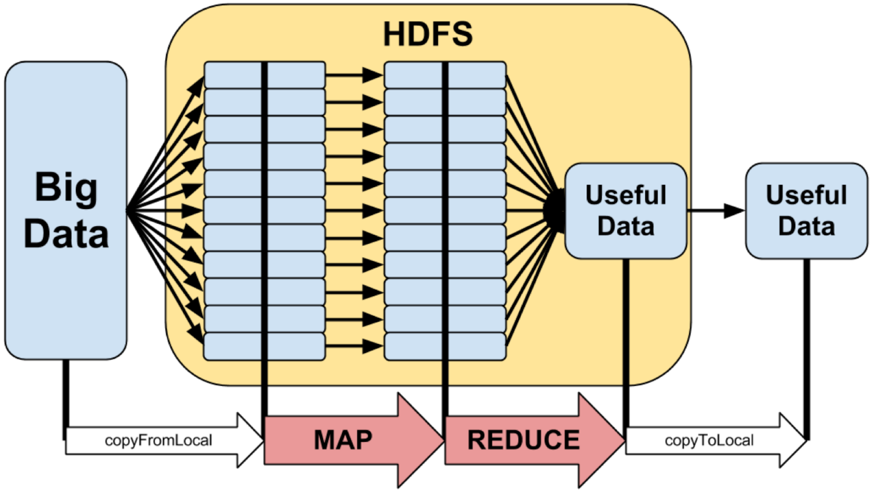
A	B	C	D
5	6	7	8
5	6	7	8
5	6	7	8

A	B	C	D
1	2	3	4
1	2	3	4
1	2	3	4
5	6	7	8
5	6	7	8
5	6	7	8

Introduction

Background of SparkR

- SparkR is the implementation of the R programming language within the Hadoop Ecosystem.
- Like other Hadoop technologies, SparkR leverages the MapReduce framework for computational power and HDFS (Hadoop Distributed File System) for storage.
- HDFS **stores data through distribution**, meaning data are partitioned and replicated throughout the Hadoop cluster for reliability and redundancy.
- MapReduce **processes data in parallel** by partitioning large data sets into manageable chunks and processing each chunk simultaneously.

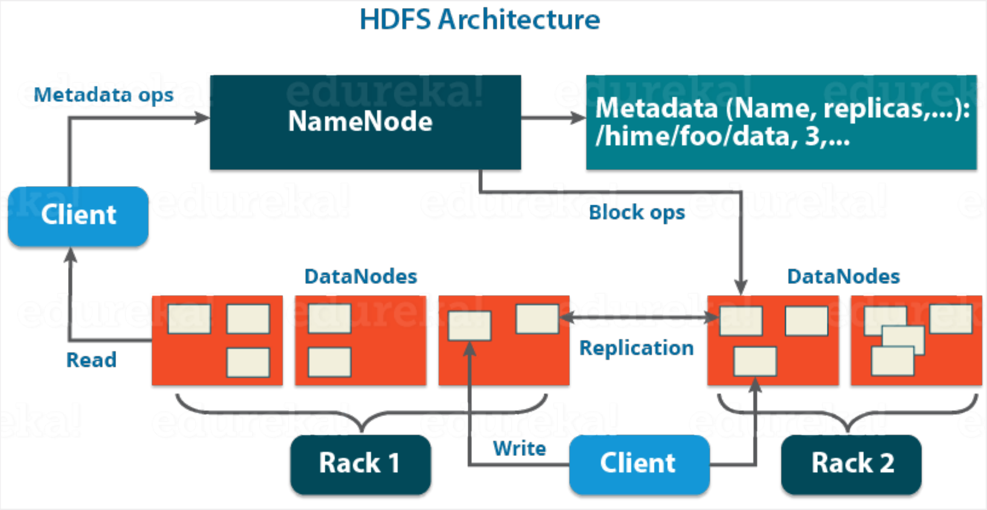


IRS

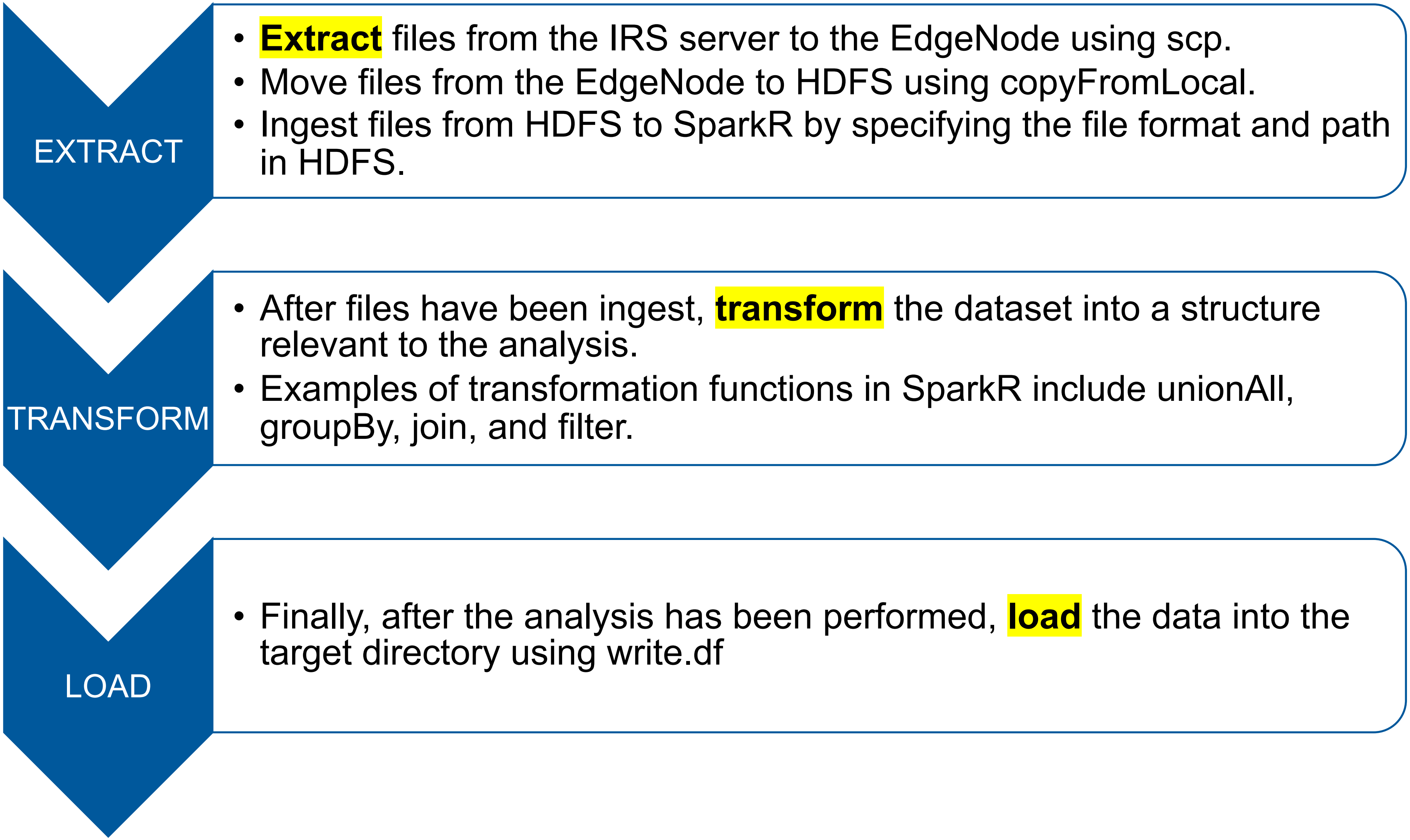
Data Organization

HDFS

- Uses a master-slave architecture where the NameNode (master) knows where the data are in the cluster while the DataNode (slave) stores the actual data.
- Data are partitioned in 128 MB blocks and distributed twice – total of three copies.
- Access HDFS using SecureCRT connected to the EdgeNode.
- Interact with HDFS using UNIX with the prefix ‘hdfs dfs’ or ‘hadoop fs’ before each command.
- ‘scp’ command is used to move files between the IRS Server and HDFS.



ETL



EXTRACT

- **Extract** files from the IRS server to the EdgeNode using scp.
- Move files from the EdgeNode to HDFS using copyFromLocal.
- Ingest files from HDFS to SparkR by specifying the file format and path in HDFS.

TRANSFORM

- After files have been ingest, **transform** the dataset into a structure relevant to the analysis.
- Examples of transformation functions in SparkR include unionAll, groupBy, join, and filter.

LOAD

- Finally, after the analysis has been performed, **load** the data into the target directory using write.df

Obstacles and Solutions

Connecting SparkR to CRAN

- **Obstacle:** IRS firewall blocked SparkR from connecting to CRAN -- the repository where R packages are downloaded.
- **Solution:** Changed the default CRAN location to the CRAN mirror located within the IRS firewall by modifying the main.R file.

Connecting SparkR to HDFS

- **Obstacle:** Unable to ingest files from HDFS due to the SparkContext environment not being created. This issue was especially tricky because the code would intermittently run with no modifications being made to the script.
- **Solution:** SparkR was loading the incorrect package version (2.3.0) instead of correct (2.4.5). Needed to specify where the correct package is located after removing earlier versions.

Ingesting .rds Files

- **Obstacle:** The team and I discovered SparkR does not have functionality for ingesting .rds files at the time of the latest release.
- **Solution:** Converted .rds file to .csv and ingested the .csv file. Not a permeant solution but a decent workaround until the functionality is added.

Thank You RAAS Team 3

- Anderson, Quinton
- Dr. Bartels, Rudy
- DiDomenico, Mario
- Lindsay, Yvonne
- Stanton, Mark

Thank you, Team 3, for your support, patience, mentorship, and humor while we worked on this project.

References

RAAS_PDC_Desk_Guide

https://spark.apache.org/docs/latest/api/R/index.html

https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html