LARGE FILES VALIDATION USING MAPREDUCE

JOSE ARMANDO FLORES DIAZ

BANK OF MEXICO

ABSTRACT

This project describes the proposed solution to solve structure validations on large files. The files that will be validated are used in the process of collecting financial information by some institutions of the financial sector.At the core of the solution is MapReduce technique.

INTRODUCTION

The financial information validation process that takes place in Bank Of Mexico, specifically in the Financial System Information Directorate, is divided mainly into two major stages. The first of them refers to a set of validations that aim to provide a certain guarantee that the file has a minimum quality in technical terms. These types of validations are called "technical validations" or structural validations. The second stage involves another set of validations which seek to identify errors in the information, directly related to the business, and this is so because it is assumed that the information does not contain technical errors, and therefore it must now look for errors in the content of the information itself. This kind of validations are called "business validations".

The validation group that this project contemplates is the one that refers to the technical validations.

TECHNICAL VALIDATION PROCESS

•The validations are applied one by one on each record in each field that comprises it.

•All validations are applied to all records.

•At the end of the validation process, it must be indicated which records contain errors, and a message indicating the validations not overcome.

•There is no dependency between records, that is, the validations are applied to each record independently and a record does not relate to what happens with another record.

Technical Validations		
Id	Name	Description
V1	Catalogued Fields	The elements of the record must belong to a previously defined.
V2	Empty	Empty values are not allowed for certain fields.
V3	Duplicates	Records with the same value for key fields are not allowed.
V4	Format (regular expression)	The elements of the record must comply with a certain format specified by a regular expression.
V5	Format (regular expression not allowed)	The elements of the record must not have any of the values defined in the regular expression not allowed.

RESEARCH POSTER PRESENTATION DESIGN © 2019
www.PosterPresentations.com

OBJECTIVES

I. Reduce through a MapReduce process, the time it takes to run technical validations to a financial information large file.

II. Show a diagnosis of the validations not exceded by the file, indicating the registration and the group of unapproved validations.

III. Evaluate the viability of a cluster of 4 commodity nodes (Banxico terminals) for the execution of a parallel programming model such as MapReduce.

GENERAL CHARACTERISTICS OF DATASET

The file to validate is of type txt. Its size is 2.3 GB, it is a file separated by "|" and it contains 91 fields.

The file that was used as an example could eventually be of a larger size since it represents information reported by financial institutions for a particular issue, but this does not exempt that in the future the issue of information is of a different nature and with a larger size.

IMPLEMENTATION

MAP-REDUCE

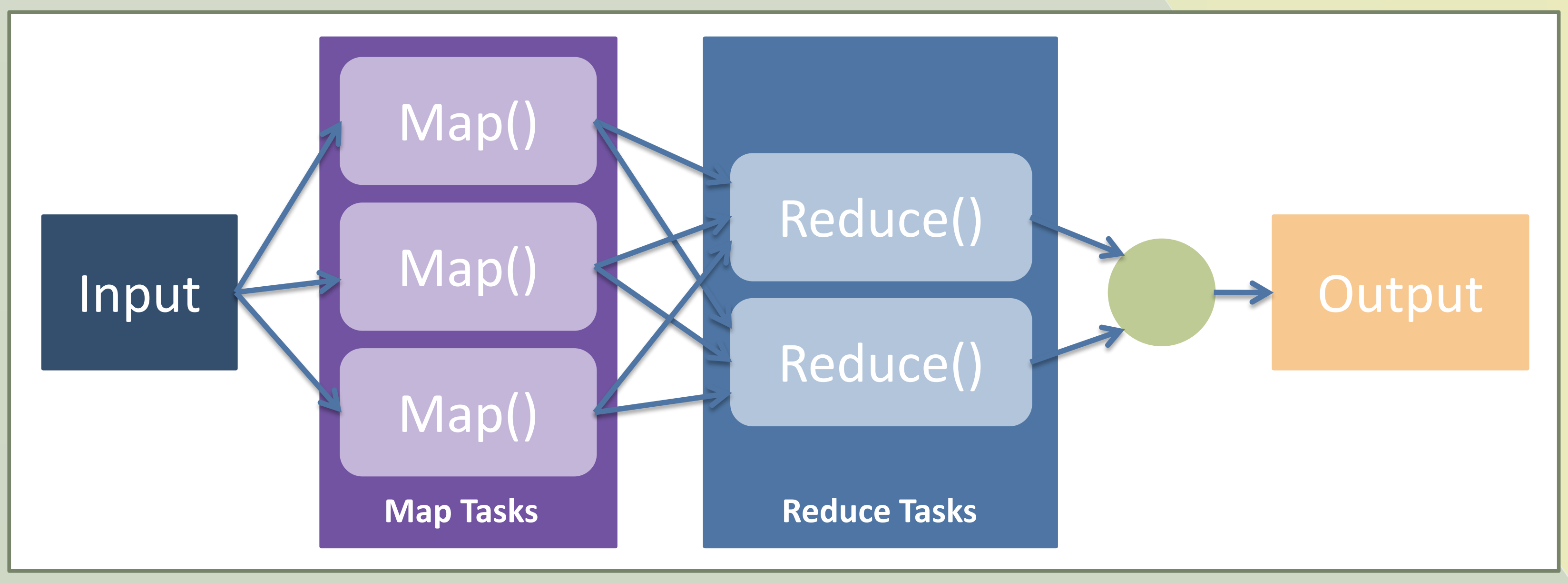
The idea behind using MapReduce is to reduce processing time. While MapReduce is not the only parallel algorithm that could support us in solving the problem, it is the one that best fits for the technical validation that we perform since it is an easily parallelizable problem due to the following factors:

•There is no dependency between records. Validations are applied to each record independently and one record is not related to what happens with another record. This allows the division of tasks and therefore its execution in parallel.

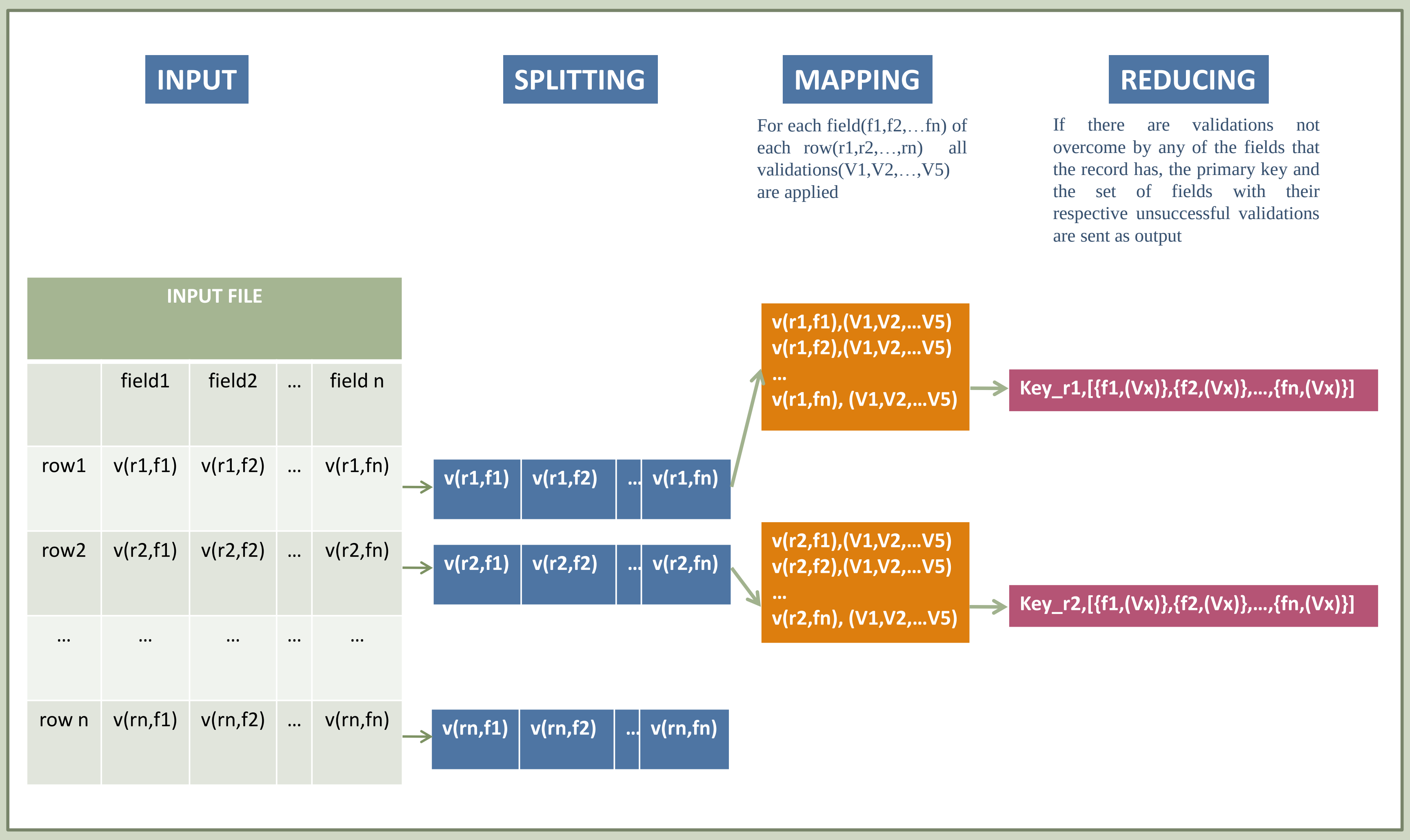
•The problem fits with the MapReduce paradigm. The record-by-record processing that must be performed on each of the records contained in the file, adjusts with the MapReduce algorithm.

•There is an intensive computation. Validations are applied one by one on each field in each record. The file that was taken as an example has 91 fields and millions of records and 5 different types of validations were applied, each with their own characteristics in terms of implementation and execution.

•All validations are applied to all records. It is important that the institutions know all the errors that their files contain, so that they can be corrected and eventually reduce the number of unsuccessful information transmissions they make.



MapReduce Overview


INPUT
SPLITTING
MAPPING
REDUCING
INPUT FILE
v(r1,f1) | v(r1,f2) | ... | v(r1,fn)
v(r2,f1) | v(r2,f2) | ... | v(r2,fn)
v(rn,f1) | v(rn,f2) | ... | v(rn,fn)
v(r1,f1),(V1,V2,...V5)
v(r1,f2),(V1,V2,...V5)
...
v(r1,fn), (V1,V2,...V5)
v(r2,f1),(V1,V2,...V5)
v(r2,f2),(V1,V2,...V5)
...
v(r2,fn), (V1,V2,...V5)
Key_r1,[{f1,(Vx)},{f2,(Vx)},...,{fn,(Vx)}]
Key_r2,[{f1,(Vx)},{f2,(Vx)},...,{fn,(Vx)}]

For each field(f1,f2,…fn) of each row(r1,r2,…,rn) all validations(V1,V2,…,V5) are applied

If there are validations not overcome by any of the fields that the record has, the primary key and the set of fields with their respective unsuccessful validations are sent as output

	field1	field2	...	field n
row1	v(r1,f1)	v(r1,f2)	...	v(r1,fn)
row2	v(r2,f1)	v(r2,f2)	...	v(r2,fn)
...	...	...	...	...
row n	v(rn,f1)	v(rn,f2)	...	v(rn,fn)

Large Files Technical Validation Implementation with MapReduce Algorithm

RESULTS

Five test events were carried out in the cluster at different times with the intention of evaluating the behavior of the cluster. For each event the following data was recorded:

•Execution date. Date on which the test was executed.

•Execution start time. Start time of the test.

•Execution time. Time it takes the execution of the algorithm in each test event.

Test events		
Execution Date	Size of the generated file [GB]	Execution Time [s]
15/07/2019	1.4	186
05/08/2019	1.4	186
07/08/2019 2:09 pm	1.4	185
07/08/2019 2:17 pm	1.4	182
20/08/2019	1.4	198
	Average	**187.4**

FUTURE WORK

Although the times obtained are aceptable, two different lines can be visualized to continue the tests with the aim of improving them. One would indicate horizontal scaling, that is, increase the number of nodes in the cluster and record the times obtained. At the same time, test larger files and verify that the benefit of using a distributed system will be more evident.

The other line is to implement a similar validation algorithm using a different technology that minimizes the disk writes that MapReduce performs, such as Spark.

CONCLUSIONS

The large file technical validation process takes approximately 7 minutes to complete under normal conditions.

With the results of this Project, a reduction of 57% was obtained (from 7 minutes it was passed to 3) under normal operating conditions.It is worth mentioning that the comparison is not quite fair in the sense that the current server where the technical validations are carried out, already has a certain load that must be resolved, so it is not possible to isolate it completely to compete on equal terms with the cluster.

Although this reduction may not be as significant, it represents a lot considering those cases where the current server is saturated.

Finally, reducing the validation time through a cluster seems to be an appropiate approach (in tests on a rented AWS cluster of 20 nodes, the time was reduced to 66 seconds). A local cluster with commodity hardware could be useful for a development environment. However, a production cluster should have more powerful nodes and enough storage space to contain all the transmitted files for a set amount of time until they are deleted.

CONTACT

jaflores@banxico.org.mx