FACULDADE DE TECNOLOGIA DE SÃO JOSÉ DOS CAMPOS

FATEC PROFESSOR Jessen Vidal

LETICIA MACEDO PRUDENTE DE CARVALHO

.

Orientador: Me. Eduardo Sakaue

São José dos Campos

2021

SUMÁRIO

[1 Introdução 3](#_Toc48568824)

[1.1 Definição do problema 3](#_Toc48568825)

[1.2 Objetivo 3](#_Toc48568826)

[2 Desenvolvimento 3](#_Toc48568827)

[2.1 Arquitetura 3](#_Toc48568828)

[2.2 Modelo de Dados 3](#_Toc48568829)

[2.3 Detalhes 4](#_Toc48568830)

[3 Resultados e Discussão 4](#_Toc48568831)

# Introdução

O SPC Brasil, é uma organização de gestão e inteligência de dados que fornece soluções de crédito, cobrança, certificação digital, antifraude, marketing e serviços financeiros. Fundada em 2002 tem como objetivo ser o elo de confiança entre as pessoas para a realização de negócios por meio de informação confiável.

A Organização possui uma estrutura de banco de dados constituída pelas Câmaras de Dirigentes Lojistas (CDL), que reúne informações do comércio de todo o país. Está presente em todo o território nacional, por meio de mais de 2.200 Entidades e mais de 1,2 milhão de Associados às Entidades.

Em virtude do ramo de atividade do SPC Brasil, os dados constituem um de seus principais insumos. Qualidade e recência são, portanto, algumas das premissas do negócio. E para atingir seus objetivos é necessário pensar limpeza, classificação e seleção dos dados para então serem utilizados nos produtos.

O fluxo responsável por construir esta Visão Única do dado foi criado com o objetivo filtrar somente informações as melhores informações de acordo com as regras de negócio e conhecimento do mercado.

Entretanto a arquitetura atual do fluxo não atende às necessidades do negócio no tempo esperado, se tornando mais demorado a cada carga e complexa para manutenção por conta da semiautomatizados dos scripts.

Devido a dificuldade descrita acima, entende-se como necessidade a compreensão do fluxo atual e a proposta de melhoria buscando diminuir o tempo do processamento, a centralização e fácil acesso às execuções de comandos ETL e arquivos de log.

## Definição do problema

Conforme descrito anteriormente, o processo atual está semiautomatizado com scripts que tornam a manutenção e acompanhamento das execuções lento e moroso e a documentação não se mostra suficiente para o entendimento o fluxo como um todo. Além disso, a arquitetura atual das tabelas impacta no aumento de tempo de leitura e escrita das cargas, gerado impacto no cluster ao excluir informações.

## Objetivo

O objetivo deste trabalho é mapear a entender o a arquitetura do fluxo atual e implementar melhorias na arquitetura do fluxo de criação do Visão Única. Esta etapa será denominada: Fusão. Espera-se otimizar o tempo de escrita e leitura das tabelas, diminuir possíveis o impacto no cluster durante a exclusão, automatizar e centralizar em uma única interface gráfica a execução dos processos e gerenciamento dos arquivos de log e documentar todo o processo.

# Desenvolvimento

A pipeline é composta por uma série de Shell-Scripts que se conectam com o ambiente Bigdata Hadoop, Spark, a ferramenta de Qualidade e Dados terceira e executam comandos de leitura, exclusão e gravação.

Cada etapa gera mensagens em arquivos de logs nas máquinas Linux quando são iniciadas, quando ocorrem erros ou quando finalizam com sucesso.

Para analisar em que condição e etapa em que cada fase se encontra é necessário se conectar à máquina Linux correspondente, caminhar até o diretório especificado no comando e analisar o arquivo de log manualmente.

A pipeline será dividida em duas fases para facilitar o entendimento e os limites deste estudo. A fase inicial será identificada como Fase de Tratamento pois é onde são feitos a higienização, correções e classificações dos dados brutos conforme são enviados pelas fontes. A segunda fase, Fusão, que será objeto de estudo deste trabalho, é onde ocorre a ordenação e classificação dos dados considerando sua qualidade e fonte de origem.

Propõe-se a criação do fluxo automatizado do Fusão para inicializar e acompanhar a execução dos comandos de forma mais simples, utilizando o software Apache Ni-Fi.

E para ganho de performance, alterar o tipo das tabelas temporárias utilizadas no processo ETL de Kudu para Parquet com particionamento a partir dos últimos dígitos do documento (CPF ou CNPJ), diminuindo assim o tempo de resposta ao excluir linhas de informações Kudu e tornando a escrita e leitura mais rápidas, já que as consultas poderão acessar diretamente a partição desejada.

Figura - Fluxo Atual de criação do Visão Única

Diagrama, Esquemático

Descrição gerada automaticamente

Conforme ilustrado na Figura 1, o fluxo atual tem como gatilho a execução manual e a execução dos comandos é semiautomatizada por Shell Scripts. O objetivo do fluxo é gerar a tabela Única Final que possui dados cadastrais, como por exemplo Dados de contato e Dados da Receita Federal, de Pessoa Física e Jurídica.

A tabela Única Final consolida as melhores informações de cada atributo cadastral, considerando as regras de qualidade de recência, ou seja, a data de atualização mais recente do dado, a qualidade da informação de acordo com as máscaras aplicadas e o nível de confiabilidade de sua origem fornecedora.

Os números de um a dez na Figura 1 representam os passos de execução consecutiva e serão detalhados abaixo.

No primeiro passo é executada uma instrução SQL que verifica quais documentos receberam atualização, através do JOIN com as “Tabelas Temporárias Atributo”, do tipo Texto.

No segundo passo o comando SQL “Delete” é executado para excluir todos os registros das “Tabelas Única Atributo”, do tipo Kudu, que retornaram na consulta do passo um.

No terceiro passo as “Tabelas Temporárias Atributos Ordenados”, do tipo Parquet, são limpas utilizando a instrução SQL “Truncate”.

No quarto passo é executado um comando SQL de leitura “Select” nas “Tabelas Temporárias Atributos”, do tipo Texto.

No quinto passo todas as linhas de informações retornadas no passo quatro são escritas nas “Tabelas Temporárias Ordenadas Atributo”, do tipo Kudu, utilizando o comando SQL “Insert”.

No sexto passo é feita a leitura das “Tabelas Temporárias Ordenadas Atributo”, do tipo Kudu, utilizando o comando SQL “Select”.

No sétimo passo a carga lida no passo anterior é dividida em lotes menores utilizando um código Spark.

No oitavo passo as informações dos lotes do passo sete são escritas nas “Tabelas Únicas Atributo”, do tipo Kudu.

No nono passo é utilizado o comando SQL “Insert” para inserir as linhas de informação nas “Tabelas Temporárias Atributo”, do tipo Kudu. Neste momento são filtrados somente cinco linhas de cada atributo, com as maiores prioridades de exibição e cada posição possui seu próprio script SQL para realizar o filtro.

No decido passo ocorre a escrita através do comando SQL “Insert” na “Tabela Única final”, do tipo Kudu. Esta tabela armazena as cinco informações com maior prioridade de cada atributo.

## Arquitetura

Figura - Proposta de Melhoria do Fluxo de criação do Visão Única

Diagrama, Esquemático

Descrição gerada automaticamente

O Apache Ni-Fi é um software que torna possível a automatização de fluxos de dados, gerenciado via interface gráfica, o que torna mais simples trabalhar com diversas origens de informações e executar processos de ETL além de possuir o agendamento de jobs e ferramentas, ou Processors, já configurados para se conectar com outras tecnologias. Portanto o software será proposto como tecnologia para a automatizar o Fluxo de construção da Visão Única e para o registro de logs será utilizada a plataforma de mensagens de fluxo de dados contínuo, Apache Kafka, que já possui conexão com o Apache Ni-Fi.

No processo de melhoria é proposto que sejam criados quatro processors no Apache Ni-Fi que farão o papel de leitura, escrita e exclusão dos dados nas tabelas temporárias e na tabela final do Visão Única. Ao total, serão somatizados dez passos de execução que terão como gatilho o agendamento da execução de acordo com a necessidade do negócio.

Conforme na pipeline atual, serão consideradas quatro tabelas de armazenamento de dados para simplificar o entendimento, denominadas: Tabelas Atributo. Entretanto, é importante ressaltar que no desenvolvimento cada atributo terá sua própria tabela, como por exemplo Tabela Telefone, Tabela Endereço e Tabela E-mail.

Cada tabela receberá as seguintes cargas.

Tabelas Temporárias Atributo: será do tipo Texto e receberá a carga de informações do delta ou ingestão recorrente;

Tabelas Temporárias Ordenadas Atributo: será do tipo Parquet, receberá a carga de arquivos das Tabelas Temporárias Atributo e a carga de dados dos documentos que estão na tabela Única Atributo e não foram atualizados.

Tabela Atributos Final: será do tipo Parquet e receberá de forma ordenada a carga de arquivos das tabelas Temporárias Atributo e Tabelas Temporárias Ordenadas Atributo. Também será particionada pelos 2 últimos dígitos do documento (CPF ou CNPJ) pois foi apontado como o método de particionamento mais balanceado e eficiente durante os testes.

Tabela Única Final tabela final: será do tipo e receberá o dado tratado e ordenado ao final do processo de delta ou ingestão, consolidando as cinco melhores informações de cada atributo

O fluxo apresenta o mesmo objetivo de construção da Visão Única, conforme descrito na sessão anterior “Desenvolvimento”, os números de um a doze na Figura 2 representam os passos de execução consecutiva feita pelos quatro processors configurados no Apache Ni-fi e serão detalhados abaixo.

No primeiro passo o processor 1 executará o comando SQL “Truncate” para eliminar todas as linhas de informações das “Tabelas Temporárias Ordenadas Atributo”.

No segundo passo o processo 2 executará a instrução SQL “Select” para ler informações das “Tabelas Temporárias Atributo”, do tipo Texto.

No terceiro passo o processor 1 irá gerar arquivos parquet a partir da leitura feita no passo dois.

No quarto passo o processor 1 irá escrever os arquivos parquet no caminho HDFS correspondente para as “Tabelas Temporárias Atributos Ordenados”, do tipo Parquet.

No quinto passo o processor 2 irá executar a instrução SQL “Select” para verificar quais documentos das “Tabelas Únicas Atributo”, do tipo Parquet, não receberam atualização, ou seja, todos os documentos que não foram encontrados nas “Tabelas Temporárias Atributo”, do tipo Texto.

No sexto passo o processor 2 irá gerar arquivos parquet a partir da leitura feita no passo cinco.

No sétimo passo o processor 2 irá escrever os arquivos parquet no caminho HDFS correspondente para as “Tabelas Temporárias Atributos Ordenados”, do tipo Parquet.

No oitavo passo o processor 3 executará o comando SQL “Truncate” para eliminar todas as linhas de informações das “Tabelas Únicas Atributo”.

No nono passo o processor 3 executará a instrução SQL “Select” para ler informações das “Tabelas Temporárias Ordenadas Atributo” ordenando as informações de acordo com a priorização da regra de negócio. Sendo a posição zero o dado com melhor classificação a e maior posição o pior dado classificado.

No décimo passo o processor 3 executará a instrução SQL “Insert” para escrever nas “Tabelas Únicas Atributo”, de forma ordenada e particionada pelo 2 últimos dígitos do documento, conforme explicado anteriormente.

No décimo primeiro o processor 4 executará a instrução SQL “Select” para ler as “Tabelas Únicas Atributo” e selecionará somente as cinco primeiras posições de cada atributo.

No décimo segundo passo o processor 4 executará a instrução de “Update” na “Tabela Única Final”, consolidando a escrita e a atualização da Visão Única.

Desta forma, é espera-se diminuir o impacto geral no cluster durante a exclusão de grandes volumes de dados Kudu, que podem ocasionar possíveis erros por tempo expirado. A mudança no tipo da tabela também busca diminuir o tempo de escrita e leitura já que os arquivos Parquet são ordenados e divididos em lotes. Cada etapa irá gerar um log com as informações padronizadas como status, tempo decorrido e id de execução e serão armazenados no Kafka para análise posterior.

## Modelo de Dados

Apresente e explique nesta Seção uma Figura ilustrando o modelo de dados utilizado.

Figura – Arquitetura do sistema



## Detalhes

Apresente nesta Seção diagramas e trechos de código importantes para o entendimento do produto desenvolvido. Apresente pelo menos um trecho de código exemplificando cada tecnologia utilizada.

Todo trecho de código deve ser comentado (use marcações ou numeração de linhas para facilitar a explicação). É altamente recomendada a utilização de listas nas explicações.

# Resultados e Discussão

Apresente neste Capítulo uma discussão sobre cada uma das tecnologias utilizadas no desenvolvimento do produto. Indique a motivação para utilização da tecnologia e como ela auxiliou no desenvolvimento. Apresente também que problemas foram encontrados e como foram solucionados.

Se há um cliente, indique se a solução foi implantada e qual a opinião dos usuários.