TCC - Bacharelado em Ciência da Computação

Detecção de fraudes pós-fato em operações cambiais baseada em análises de grafos

Proposta

IME-USP

Victor Oreliana Fernandes Faria

Universidade de São Paulo

Brasil - 04 / 2019

1 - **Introdução**

1.1 **Orientação**

Orientação: Profa. Dra. Kelly Rosa Braghetto - IME USP

Co-orientação: Márcio Willian Caldeira de Melo, CTO - BeeTech Global

Coordenação geral: Profa. Dra. Nina Sumiko Tomita Hirata - IME USP

1.2 **Contexto**

Segundo a atual normativa do Banco Central do Brasil (BACEN) a respeito do mercado de câmbio [1], cabe às instituições privadas zelar pela conformidade de remessas de capital ao exterior. Esta regulação traz para as instituições cambiais a responsabilidade de desenvolver métodos analíticos que processem informações de remessas relativas ao agente remetente, ao agente recebedor e à transação em si. Assim, essas instituições podem atuar como órgão regulador que fiscaliza a transação sobre os aspectos tributário e criminal,enviando relatórios de análise fiscal aos devidos órgãos públicos responsáveis. Tal fiscalização é fundamental para: (i) dificultar e prevenir o financiamento de organizações criminosas - OLIVEIRA[2] mostra a relevância do câmbio no âmbito criminal; e (ii) para reduzir a sonegação tributária - são arrecadados bilhões de reais em tais operações anualmente [3] e portanto mesmo um pequeno aumento percentual é bastante significativo.

A análise de conformidade é tipificada em duas categorias, pré-fato e pós-fato. A primeira categoria contempla a identificação de suspeitas de fraude de forma tal que as informações de uma única remessa são necessárias e suficientes para apontá-la como suspeita. A segunda contempla a identificação de suspeitas de fraude através da análise da relação entre conjuntos de atributos de duas ou mais remessas. Análises pré-fato são computacionalmente simples porque podem ser realizadas confrontando um conjunto de regras estruturadas sobre uma

modelagem relacional de dados ao dado de entrada - e.g. uma pessoa física realizando uma remessa para uma pessoa jurídica declarando estar fazendo-a para outra pessoa física a fim de se extinguir de tributações comerciais. No entanto, as análises pós-fato são computacionalmente interessantes.

Podemos exemplificar, brevemente, dois casos de análise pós-fato computacionalmente interessantes as quais este trabalho visa contemplar:

1. O caso de lavagem de dinheiro através de co-autores; dada a existência de uma quantidade de recursos ilícitos a serem enviados para o exterior, esta pode ser remetida por um número, N, de co-autores sem precedentes criminais. Nesse caso, o valor total a ser remetido é dividido entre os co-autores de forma que cada um realize uma remessa de baixo valor, aparentemente em conformidade (remessas de baixo valor possuem menos critérios de fiscalização e para o BACEN baixo valor é dado como qualquer valor até o equivalente em reais a 3 mil dólares americanos). Sendo T o número de linhas de uma tabela de transações em um banco de dados relacional, cada busca por suspeitas de fraudes deste tipo irá requerer a varredura de toda a tabela resultando em uma complexidade O(T) e portanto para a emissão de um relatório contemplando todos os casos possíveis resultará em uma complexidade $O(T^2)$, restando ainda a implementação de regras para retirar repetições.

2. O caso de sonegação fiscal através de um anel de fraude. Um anel de fraude é definido por um grupo de agentes operando em conjunto entre si, individualmente dentro da normalidade, porém cujas interações resultam em fraude. Por exemplo, considere um caso em que 3 pessoas (A, B e C) realizam remessas de igual valor de forma que A envia para B, B para C e C para A. Esse caso é equivalente a cada um dos agentes enviar uma remessa para si, entretanto o regime tributário do primeiro caso resulta ganho de 289.47% em impostos não arrecadados quando comparado ao segundo - a alíquota de IOF (Imposto sobre Operações Financeiras) no primeiro caso é de 1.1% e no segundo de 0.38%. Convencionalmente, em um banco de dados relacional, uma consulta para revelar tais anéis de fraude consumiria tempo O(N^T), onde N é o tamanho do anel e T é o número de linhas da tabela de transações, por ser

necessário realizar um produto cartesiano (TxT) a cada iteração i (1 <= i <= N) que compõe a identificação do anel. Além disto, como no caso anterior, restaria a implementação de regras para remover repetições.

A fim de lidar efetivamente com esses problemas computacionais, propomos uma abordagem de processamento embasada em bancos de dados orientados a grafos cujo desempenho teórico do tempo de consulta é O(T). De fato, encontramos casos de sucesso na literatura que são similares a respeito de modelagens de fraudes com grafos [4][5].

1.3 **Problema**

Estudaremos neste trabalho o caso de fraudes pós-fato de operações de câmbio no Brasil cujos remetentes são pessoas físicas, nos restringindo à identificação e modelagem de padrões de grafos e de um sistema de pontuação sobre as entidades. Os grafos deverão envolver pessoas e transações e o sistema de pontuação diz respeito ao cálculo de uma média ponderada de uma pontuação atribuída a padrões de grafos encontrados e a indicadores de estatística descritiva entre os agentes; esta média será usada para classificar o risco potencial de uma transação de acordo com uma escala configurável.

1.4 **Objetivos**

1. Desenvolver uma modelagem de grafos na qual seja possível a identificação padrões que contemplem fraudes de câmbio.

2. Desenvolver um sistema de monitoramento de transações computacionalmente eficiente que seja capaz de identificar padrões de fraudes cambiais pós-fato e realizar classificações de risco.

3. Criar uma coleção de dados fictícia que permita aferir que o sistema realmente é capaz de detectar as fraudes propostas. Esta coleção deve estar acompanhada de uma documentação anexa detalhando a reprodutibilidade do experimento.

4. Apresentar uma comparação de desempenho computacional entre uma busca de uma fraude em um banco de dados relacional e outra em um banco de dados orientado a grafos.

5. Realizar um estudo de caso e obter ou uma validação da resolução do problema para o caso proposto ou uma análise apontando o que deve ainda ser desenvolvido para atingir a resolução do problema no caso em particular.

2. **Métodos**

2.1 **Desenvolvimento**

Detalhamos aqui as etapas de desenvolvimento, sequencialmente:

1. Identificação dos padrões transacionais associadas às fraudes de interesse.
2. Modelagem de grafos respectivos aos padrões citados no item anterior.
3. Modelagem do banco de dados orientado a grafos.
4. Definição do mapeamento entre os bancos de dados relacional e orientado a grafos.
5. Especificação de consultas em linguagem Cypher para identificar os padrões do item (1) na modelagem do item (3).
6. Estudo de caso (seção 2.3) - desenvolvimento e implantação de um sistema de monitoramento das fraudes propostas em remessas.
7. Elaboração da análise dos resultados do item (6)
8. Análise comparativa entre o desempenho computacional das consultas SQL vs Cypher em seus respectivos SGBDs e modelagens.

2.2 **Ferramentas**

2.2.1 **Desenvolvimento do Sistema**

O sistema tratar-se-á de um sistema web (cliente-servidor) estabelecido sobre a especificação arquitetural para RESTful-APIs utilizando atentamente os recursos oferecidos pelo protocolo HTTP 1.1 e seguindo a abordagem de DDD (*domain driven design*) a fim de permitir futuras expansões de complexidade sistêmica com baixo custo de manutenção da engenharia do software. Seu desenvolvimento será integralmente em JavaScript rodando sobre a Node.js® runtime. Para persistência de dados, será utilizada uma abordagem poliglota combinando banco de dados relacional e banco de dados orientado a grafos, conforme os respectivos SGBDs: MariaDB e Neo4J. A infraestrutura será fundamentada no serviços de nuvem da Amazon (*Amazon Web Services* - AWS) e containers Docker, utilizando mecanismos de CI/CD (*continuous integration and deployment*) e controle de versões a serem definidos a posteriori todavia que permitam uma fácil reprodutibilidade do ambiente de execução dos experimentos.

Todas ferramentas a serem utilizadas na composição do sistema e em sua implantação possuem licenças de software permissivas a fins acadêmicos, bem como todo o desenvolvimento do trabalho será realizado priorizando o uso softwares livres.

2.2.2 **Visualização de Resultados**

Serão compilados relatórios com gráficos a respeito das modelagens e das análises comparativas. Para tanto, usaremos o suporte nativo do Neo4J para visualização de grafos e o Seaborn Pydata para demais gráficos.

2.3 **Estudo de Caso**

2.3.1 **Parceria Acadêmica**

O estudo de caso a ser conduzido será sobre a plataforma RemessaOnline® e ocorrerá com a supervisão do co-orientador deste trabalho, co-fundador da plataforma e especialista em prevenção a lavagem de dinheiro, Márcio Willian Caldeira. Nesta data, a plataforma já transacionou mais de 7 bilhões de reais e foi utilizada por mais de 150 mil clientes.

2.3.2 **Condução e Desenvolvimento do Experimento**

O estudo será feito partindo da modelagem relacional existente na plataforma no ambiente de produção, então será construído um programa de mapeamento dos dados relacionais para a modelagem de grafos proposta . Visando à privacidade dos usuários envolvidos, o programa mapeador deve encriptar dados sensíveis. A saída deste programa será uma série de requisições HTTP POST à API do sistema de monitoramento, que deve se encarregar de, a partir de uma representação normalizada intermediária, realizar a inserção dos dados no modelo de grafos proposto. Ter uma representação intermediária dos dados é importante para ficar estabelecido um contrato de API estável que não mude ao longo do tempo ou ao menos que seja retrocompatível independentemente da modelagem de dados intrínseca ao sistema de monitoramento. Também haverá a atenção à questão de disponibilidade do sistema, ou seja, serão aplicadas técnicas de computação paralela e de engenharia de software que contemplem de fato uma real utilização do sistema de produção em uma arquitetura de serviços web independentes.

2.3.3 **Avaliação**

A RemessaOnline® conta com uma série de processos de conformidade, nos quais são envolvidos diversas ferramentas de apoio à tomada de decisão e especialistas. Porém, a respeito de análises pós-fato, o processo de análise é majoritariamente manual e apoiado em planilhas - o que é uma realidade não só desta plataforma mas como do setor de câmbio em geral, o que é devido à ausência de ferramentas específicas para este fim. No entanto, apesar das limitações do cenário atual, algumas fraudes são detectadas. Por meio das fraudes já catalogadas, será possível fazer uma avaliação de desempenho do sistema que será desenvolvido neste trabalho.

3. **Planejamento e Cronograma**

3.1 **Elementos do Planejamento:**

1. Ideação do projeto.
2. Estudos sobre Neo4J.
3. Elaboração da proposta.
4. Elaboração de modelos de dados: modelagem de grafos e modelagem relacional.
5. Desenvolvimento do sistema.
6. Desenvolvimento do Importador de Dados.
7. Implantação de infraestrutura / Coleta de dados.
8. Criação do conjunto de dados para reprodutibilidade.
9. Desenvolvimento do pôster.
10. Desenvolvimento da monografia.
11. Elaboração da apresentação da análise dos resultados.

3.2 **Cronograma**

Elemento / Mês	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
1	■	■	■									
2			■	■								
3				■								
4			■	■	■							
5				■	■	■	■					
6							■	■				
7							■	■	■			
8									■			
9									■	■		
10							■	■	■	■	■	
11										■	■	

Tabela 1- Cronograma. Índice do elemento de planejamento vide seção 3.1 por mês do ano de 2019. Células em preto significam que é esperado que o elemento de planejamento respectivo seja desenvolvido durante o mês da respectiva coluna.

4. **Glossário Técnico**

- AWS - Plataforma de Computação em Nuvem da Amazon Inc.
- CI/CD - Metodologias de engenharia de software que automatizam o processo de deploy e guiam um desenvolvimento estável de software sob a ótica de operações [6][7].
- Cypher - Linguagem de consulta declarativa de grafos [8].
- DDD - Padrão arquitetural de software [9]
- Docker - Software para virtualização em nível de sistemas operacionais [10].
- HTTP 1.1 - Hypertext Transfer Protocol - https://www.w3.org/Protocols/rfc2616/rfc2616.html.
- HTTP POST - método HTTP - https://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html#sec9.5.
- JavaScript - Linguagem interpretada de programação de alto nível
- SGBD - Sistema Gerenciador de Banco de Dados.
- SQL - Linguagem de Consulta Estruturada para bancos de dados relacionais
- Maria DB - SGBD Relacional
- Neo4J - SGBD Orientado a Grafos
- NodeJS - Sistema de Execução JavaScript mantido pela Linux Foundation
- RESTful API - Padrão arquitetural para desenvolvimento de *web services* baseado em Representational State Transfer - https://en.wikipedia.org/wiki/Representational_state_transfer.
- Seaborn Pydata - Software para visualização de dados em linguagem Python - https://seaborn.pydata.org/.

5. **Referências**

[1] BACEN. Resolução n° 2554, de 24 de setembro de 1998. Dispõe sobre a implantação e implementação de sistema de controles internos.

[2] OLIVEIRA, Luís Flávio Zampronha de. **REMESSAS DE CAPITAIS AO EXTERIOR: a lavagem de dinheiro através da evasão de divisas**. 2012. 72 f. Monografia para obtenção do título de Especialista em Gestão de Política de Segurança Pública da Academia Nacional de Polícia.

[3] Secretaria da Receita Federal do Brasil, Centro de Estudos Tributários e Aduaneiros. **Análise da Arrecadação das Receitas Federais**. Maio de 2018.

[4] RALHA, Cecília et al, **Banco de Dados em Grafo: Um Estudo de Caso em Detecção de Fraudes no Governo Brasileiro**. 2017. Universidade de Brasília.

[5] HOLANDA, Maristela; ARAUJO, Gabriel M. **Uso de banco de dados orientado a grafos na detecção de fraudes nas cotas para exercício da atividade parlamentar**. 2018 Universidade de Brasília.

[6] FOWLER, Martin. **Continuous Integration**, disponível em <https://martinfowler.com/articles/continuousIntegration.html>, acessado em Abril de 2019.

[7] FOWLER, Martin. **Continuous Delivery**, disponível em <https://martinfowler.com/bliki/ContinuousDelivery.html>, acessado em Abril de 2019.

[8] NADIME, Francis; Alastair Green; Paolo Guagliardo, Leonid Libkin, Tobias Lindaaker, et al.. **Cypher: An Evolving Query Language for Property Graphs**. SIGMOD'18 Proceedings of the 2018 International Conference on Management of Data. 2018, pp.1433.

[9] EVANS, Eric. **Domain-Driven Design: Tackling Complexity in the Heart of Software**. Addison-Wesley. ISBN 978-032-112521-7.

[10] HYKES, Solomon. **Docker (Software)**, disponível em <https://en.wikipedia.org/wiki/Docker_(software)>, acesso em Abril de 2019.