MINERAÇÃO DE DADOS WEB COM PROCESSOS ETL EM BASH SCRIPT

Vinícius Silva Madureira Pereira[1], Fábio Luiz Viana[2]

RESUMO: Este trabalho propõe meios para realizar mineração de dados, identificando padrões e captando informações *Web* ao utilizar processos de extração, transformação e carregamento de dados (ETL) de forma sinóptica e eficiente, empregando programas de código-fonte aberto e que partilham da filosofia *Free Software*. Tenciona-se, ainda, demonstrar como ferramentas triviais e que, em distribuições do ambiente GNU/Linux, raramente necessitam de instalação ou, até mesmo, uma interface gráfica – como o Bash, roboram demasiada performance tanto em tempo quanto em custos.

PALAVRAS-CHAVE: Mineração de Dados, Processos ETL, Bash.

INTRODUÇÃO

Nos atuais cenários, acadêmico e profissional, ao abordar a temática mineração de dados remete, amiúde, a algoritmos excessivamente compostos e com prolixas análises estatísticas, exprimindo ao aluno ou ao funcionário inexperiente uma curva de aprendizado íngreme, fastidiosa e, geralmente, improvável de se alcançar, principalmente em consequência da escassez de prazos, vistas à azáfama cotidiana.

Urge, então, a necessidade de métodos ágeis e simplificados para introduzir tais indivíduos no contexto da prospecção de dados, propondo modelos inteligíveis, com níveis apropriados de aprendizado e aptos a servirem de substrato para um novo grau de abstração, abrangendo determinados domínios – como a *Web*.

OBJETIVOS

A mineração de dados *Web* com processos ETL em Bash Script objetiva demonstrar o desenvolvimento e a execução de um sistema simples que, por meio da assimilação de padrões unívocos abstraídos de um domínio específico, armazena endereços de *Proxies Servers*.

MATERIAIS E MÉTODOS

Análise de referencial teórico, cuja abordagem enfatiza o desenvolvimento de algoritmos em Bash Script e Expressões Regulares (RegExp) a fim de gerenciar parâmetros, além da integração de programas de linha de comando como: WGet, cURL, GREP, AWK, Sed e SQLite.

Outrossim, foram implementados modelos de diagramas UML (Linguagem de Modelagem Unificada) e ER (Entidade-Relacionamento) para facilitar, tanto aos apreciadores do âmbito quanto aos leigos, o entendimento do processo de desenvolvimento de *software* do projeto em questão.

A metodologia ETL (*Extract, Transform and Load* – Extrair, Transformar e Carregar), embora amplamente difundia, quase que exclusivamente, no contexto de *Data Warehouse* (Armazém de Dados), revelou-se significativamente profícua à mineração de dados. Uma vez que ela parte de premissas básicas como identificar e assimilar padrões para ulterior processo de sintetização dos dados e persistência em uma base, essa sistematização adequou-se com excelência à tarefa de minerar *proxies*.

RESULTADOS E DISCUSSÃO

Em testes realizados foram capturados e persistidos na base de dados 470 *proxies* em 2 minutos e 30 segundos (150 segundos), ou seja, cerca de 1 *proxy* a cada 0,32 segundos, o que é um valor consideravelmente viável para uma linguagem interpretada como o Bash Script (foi utilizada uma conexão de 25Mbps).

[1] Estudante do curso Tecnologia em Análise e Desenvolvimento de Sistemas, IFSP Campus Barretos, Av. C-1, 250, CEP 14.781-502, Barretos, SP. Bacharel em Sistemas de Informação (UNIFEB) e professor de nível técnico (Eletrônica e Automação Industrial – LiceuTec), vinicius.madureira@aluno.ifsp.edu.br.
[2] Orientador, Professor do curso Tecnologia em Análise e Desenvolvimento de Sistemas, IFSP Campus Barretos, Av. C-1, 250, CEP 14781-502, Barretos, SP, fabio.viana@ifsp.edu.br.

A Figura 1, como pode ser observada a seguir, demonstra uma consulta SQL que resulta nos *proxies* dos países: Alemanha, Brasil, China e França.

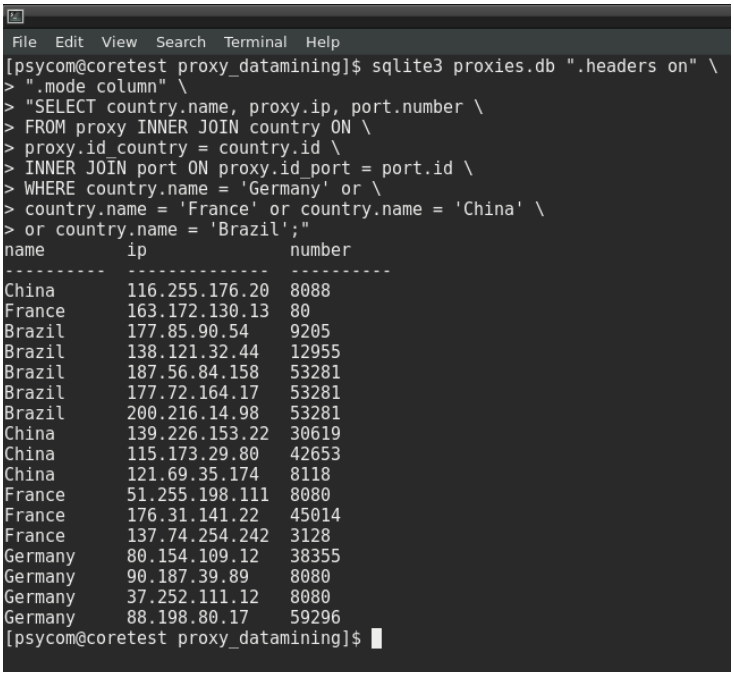

Figura 1: Consulta SQL com os *proxies* capturados

Após configurar o navegador QupZilla, que estava inicialmente configurado com um IP de Internet atribuído pelo DHCP do ISP (*Internet Server Provider*) – 186.195.210.48, com o último *proxy* da Figura 1, ou seja, 88.198.80.17:59296 – pertencente à Alemanha, o IP foi alterado para 176.9.47.248, pois, uma vez anônimo, o servidor pode trocar de valor em determinados intervalos de tempo. A própria página da Google Inc., já com o *proxy* configurado, não foi exibida em alemão, mas em russo.

CONCLUSÕES

Embora as linguagens compiladas possam ser executadas até 100 vezes mais rápidas do que as interpretadas (SEBESTA, 2006), a fim de eficiência para desenvolver algoritmos que demandam testes de similaridades e padrões, as linguagens interpretadas são sobremodo muito mais vantajosas.

AGRADECIMENTOS

Dedico este trabalho, primeiramente, à Santíssima Trindade dos atualmente denominados cristãos, composta pelo Deus Pai, pelo Deus Filho e pelo santíssimo Espírito Santo de Deus. Posteriormente, tributo-o à minha mãe, Odete Ferreira da Silva, a qual dedicou e ainda dedica a sua vida para que as minhas realizações se tornem possíveis. Agradeço também ao orientador desse artigo, Prof. Me. Fábio Luiz Viana, pelo apoio e valiosas correções.

BIBLIOGRAFIA

NEGUS, C. **Linux a Bíblia. O Mais Abrangente e Definitivo Guia Sobre Linux**. 1 ed. Rio de Janeiro: Alta Books, 2014.
CEZAR, J. **Programação Shell Linux**. 10 ed. São Paulo: Brasport, 2014.
PASSOS, E. **Datamining. Conceitos, Técnicas, Algoritmos, Orientações e Aplicações**. 2 ed. Rio de Janeiro: Campus, 2015.
SEBESTA, R. W. **Conceitos de linguagens de programação**. 5. ed. Porto Alegre: Bookman, 2006.