

Mineração de Dados Web com Processos ETL em BASH Script

```
do
02d" $page)
##### Page: $page #####
l --silent "http://nntime.com/proxy-list-$page.htm" | grep --null-data --text --only-matching '<tr cl
o -e "$table_rows" | sed 's/<tr /<n<tr /g')

e_rows; do
$row" | sed 's/<td/<n<td/g')
$echo "$row" | sed '3p; d' | grep --perl-regexp --only-match '(\w+)' | sed 's/\+//g' | awk '{print
$row" | sed '2p; d' | grep --perl-regexp --only-match '(?<=\\)(\\d+)(\\.)?)+' | awk '{print
$row" | sed --expression '4p; d;' | sed 's/<\\?td>/g')
ho "$row" | sed '5p; d' | grep --perl-regexp --only-match '\\w{3}-\\d{1,2}-\\d{4}'" "$echo "$row
"$row" | sed '6p; d' | sed --expression 's/<\\?td>/g' --expression 's/ \\?\\.*/g')
"$row" | sed '7p; d' | sed --expression 's/<\\?td>/g' --expression 's/ \\?\\.*/g')
]=$ip ["port"]=$port ["type"]=$type ["country"]=$country ["owner"]=$owner

{
ho ${proxy["ip"]} | cut --characters=1-15)
echo "${proxy["port"]} | cut --characters=1-5)
te3 proxies.db "SELECT ip as proxy_ip, id_port as proxy_port FROM proxy INNER JOIN port ON proxy.id_po
tset" ]]; then
]=$echo "${proxy["type"]} | sed --expression='s/ \\?proxy//g' | cut --characters=1-20)
update"=$(date --date "${proxy["last_update"]} " +%Y-%m-%d %H:%M:%S")
y["country"])= " , " ]]; then
ountry"=$(echo $country | grep --perl-regexp --only-matching '(?U).*(?=,)' | cut --characters=1-30)
")=$(echo "${proxy["owner"]} | cut --characters=1-80)

-----Trying to insert proxy:-----\\n
Port: ${proxy["port"]}\\nType: ${proxy["type"]}\\nCountry: ${proxy["country"]}\\nOwner: ${proxy["owner"]}
"]="proxy")
t next id "${dataset["table"]}"
"]="port" ["column"]="number" ["signal"]="=" ["value"]=${proxy["port"]}
get id)
"]="type" ["column"]="name" ["signal"]="=" ["value"]=${proxy["type"]}
get id)
"]="country" ["column"]="name" ["signal"]="LIKE" ["value"]=${proxy["country"]} | sed --expres
=$(get id)
"]="owner" ["column"]="name" ["signal"]="=" ["value"]=${proxy["owner"]} | sed --expression="s
```

Contexto

- O ensino da Mineração de dados
- Processos ETL
- Os benefícios das expressões regulares em relação ao custo do tempo de processamento
- Os constantes avanços do BASH

Vantagens

- Testes unitários a cada comando inserido
- Utilização de recursos do sistema operacional sem a necessidade de importação de bibliotecas
- Utilização de programas como recursos de linguagem
- Acesso à base de dados diretamente do *script*

Desvantagens

- Alto consumo de memória e processamento
- Conhecimento considerável acerca de inúmeras ferramentas (programas e comandos)
- Não existe, ainda, uma padronização de desenvolvimento
- A Web é altamente dinâmica e o algoritmo pode depender de serviços de terceiros

Projeção

- Desenvolvimento de uma API para *scripts*
- Maturação de projetos em conjunto com aplicativos CMS
- Implementar, em conjunto com a Free Software Foundation (Projeto GNU) por meio da plataforma Savannah, uma biblioteca ETL

Referências

- JARGAS, A. M. Shell Script Profissional. 1 ed. São Paulo: Novatec, 2008.
- NEGUS, C. Linux a Bíblia. O Mais Abrangente e Definitivo Guia Sobre Linux. 1 ed. Rio de Janeiro: Alta Books, 2014.
- SILVA, L. C.; FERRAR, D. G; QUERO, P. Introdução à Mineração de Dados. Conceitos Básicos, Algoritmos e Aplicações. 1 ed. São Paulo: Saraiva, 2016.
- CEZAR, J. Programação Shell Linux. 10 ed. São Paulo: Brasport, 2014.

Referências

- MORIMOTO, C. E. kurumin 7. Guia Prático. 1 ed. Paraná: Sulina, 2007.
- 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. BEZERRA, E. Princípios de Análise e Projeto de Sistemas com UML. 2. ed. Rio de Janeiro: Elsevier, 2007. 369 p.

Dúvidas



Agradecimentos...

Muito obrigado!!!