Recuperação de Informação na Web 2017/2

Construtor de um Coletor de Propósito Geral – 10 pontos

Com o objetivo de estudar e aprender a arquitetura de um coletor simples para Web, você deverá fazer um coletor de propósito geral além de fazer uma análise do impacto dos parâmetros deste coletor.

Data da entrega: 02 de outubro as 23:59. Tolerância máxima de 6 horas.
Não será aceito novos integrantes após a entrega do trabalho, mesmo se estiver "esquecido" de colocar.

Sobre a 'Equipe'

O projeto deve ser desenvolvido por grupos de no máximo quatro pessoas. Caso o grupo exceda este valor, ele perderá 2 pontos por pessoa excedente. O grupo sera definido no dia da pratica avaliativa

Pratica Avaliativa

Com o objetivo de iniciar a codificação, serão realizadas praticas avaliativas no formato de Coding Dojo. A presença é obrigatória de todos os integrantes do grupo. Integrante que faltoso perdera um ponto de cada pratica, em caso de atraso, ele perdera de 0,5 ponto (30 minutos de atraso) ate 1 ponto (a partir de 30 minutos de atraso).

Tarefas

Para fazer um coletor é necessário:

1) O escalonador deverá possuir:
   (a) Uma fila de páginas, sendo coletado através de uma busca em largura
   (b) Não permitir que a mesma url seja coletada mais de uma vez
   (c) Armazenar a última vez que um servidor foi acessado. Pois, um servidor poderá ser acessado de 30 em 30 segundos
2) Multiplas threads para coletar as páginas (os *Page Fetchers*):
   (a) Coleta a página que o escalonador organizou
   (b) Dado a página coletada, extrair seus links e inserir na fila do escalonador todas as páginas coletadas
      → A classe ColetorUtil poderá auxiliar no caso de urls relativas, pois os links devem ser sempre adicionados no seu formato completo na fila
   (c) Levar em consideração páginas html mal-formadas. Você poderá usar uma API para isso.
   (d) Levar em consideração o encoding da página. A classe ColetorUtil pode auxiliar neste processo
   (e) Caso a página não exista, a mesma é ignorada

Você deverá fazer um coletor que obedeça no mínimo os seguintes requisitos:

1) Obedecer os protocolos de exclusão de robôs:
   (a) critérios pertencentes no robots.txt
      - Pode ser usado uma API para isto
   (b) Critérios "noindex" e "nofolow" das metatags de cada html extraído
   ps: noindex: não é permitido coletar. Nofolow: não é permitido seguir os links por meio desta página
   (c) Obedecer o prazo de, no mínimo, 30 segundos entre requisições em um mesmo servidor (*hostname*)

→ **Caso não obedeça esses critérios o grupo perderá 7 pontos**

2) Criar um nome no "*User agent*" (finalizando como bot) e uma página pessoal com descrição do coletor, nome dos membros dos grupos, datas das coletas e propósito das coletas além de um e-mail de contato. Deverá ser explicitado que este coletor baixa

apenas páginas públicas sempre levando em consideração a política de exclusão de robôs (robots.txt). Esta página deverá ficar online durante o semestre todo. O endereço da página pessoal deverá ser definida no User agent. Exemplo: “meuBot (wordpress.org/infoMeuBot)”.

→ **O grupo perderá 7 pontos** caso não crie a página e/ou não utilize o nome no “User Agent” devidamente e de acordo com o especificado acima

→ **Exemplo de páginas de informação de robôs:**

https://support.apple.com/en-us/HT204683

2) Utilizar os seguintes parâmetros no coletor:
   → Número máximo de páginas (500 páginas)
   → Profundidade por domínio (4 páginas)
   → Número de *threads* utilizado
3) Utilizar as sementes de acordo com os integrantes do grupo, usando tabela de sementes que está apresentado no Moodle em documento separado
4) Produzir um relatório, a ser definido na próxima seção
5) Armazenar a lista de URLs coletadas

Conteúdo do Relatório

O relatório é uma parte importante do trabalho e será levado muito em consideração. O relatório deve possuir os seguintes tópicos:

a) Principais desafios, decisões e arquitetura utilizada
b) URLs sementes utilizadas
c) Como foi feito, faça referencias à classes e métodos do código fonte:
   → Os critérios de exclusão de robôs e quantidade de tempo entre requisições à um mesmo servidor
d) O impacto na velocidade de coleta (quantidade de páginas por segundo) ao aumentar o número de threads 10 a 100, de 20 em 20
e) Link para a página descrevendo o coletor criado

O código fonte deverá vir separadamente e não dentro do relatório

Bibliotecas utilizadas

Vocês poderão usar bibliotecas para os seguintes propósitos (segue também algumas sugestões de APIs)

→ Extração dos links de páginas HTML mal-formadas:
   - HTMLCleaner (htmlcleaner.sourceforge.net)
→ Parser do protocolo de exclusão de robôs
   - jrobotx (github.com/TrigonicSolutions/jrobotx)
      Dependências: apache commons io e log4j
→ Classe ColetorUtil
   - Dependencias: juniversalchardet

Entrega

A entrega deverá conter os seguintes itens em um arquivo comprimido:

a) Relatório do trabalho **obrigatoriamente em PDF**
b) Código fonte
c) Lista de URLs coletadas

Critério de avaliação

O trabalho será avaliado de acordo com:

1) Legibilidade, comentários e organização do código
2) Funcionamento do coletor

3) Uso do protocolo de exclusão de robôs
4) Página de informação do coletor
5) Lista das páginas coletadas
6) Conteúdo do relatório com todos os itens requisitados

Caso não seja feito (ou feito incorretamente) o item (3) e/ou (4) o grupo perderá 7 pontos. O conteúdo do relatório é tão importante quanto o funcionamento do coletor. O grupo poderá perder até 75% dos pontos caso seja produzido um relatório mal escrito ou incompleto

Plágio não será tolerado e, caso identificado, o grupo (quem forneceu e quem utilizou) terá seu trabalho zerado.

Política de atraso

Será descontado 1 ponto por dia de atraso.

Dicas Protocolo Exclusão de Robôs

Usando a API sugerida, se você fizer isto:

```
RobotExclusion robotExclusion = new RobotExclusion();

Record rFB = robotExclusion.get(new URL("https://www.facebook.com/robots.txt"),"daniBot");
System.out.println("Aceitou o fb index?  "+rFB.allows("/index.html"));
System.out.println("Aceitou o fb o cgi-bin?  "+rFB.allows("/cgi-bin/oioi"));
System.out.println("Aceitou o fb o oioi?  "+rFB.allows("/lala/oioi"));

Record rTerra = robotExclusion.get(new URL("http://www.terra.com.br/robots.txt"),"daniBot");
System.out.println("Aceitou o terra index?  "+rTerra.allows("/index.html"));
System.out.println("Aceitou o terra o cgi-bin?  "+rTerra.allows("/cgi-bin/oioi"));
System.out.println("Aceitou o terra o oioi?  "+rTerra.allows("/lala/oioi"));
```

A saída esperada será:

```
<terminated> EscalonadorSimples [Java Applic
Aceitou o fb index?  false
Aceitou o fb o cgi-bin?  false
Aceitou o fb o oioi?  false
Aceitou o terra index?  true
Aceitou o terra o cgi-bin?  false
Aceitou o terra o oioi?  true
```

Atenção: Cada vez que você chamar o “get” da classe RobotExclusion, você fará uma requisição no robots.txt. Faça o código de tal forma que você precise fazer apenas uma requisição por robots.txt. Lembre-se que requisições Web são muito demoradas.