

# Trabalho Prático 1 - Recuperação da Informação

Matheus Aquino Motta<sup>1</sup>

<sup>1</sup>Bacharelado em Matemática Computacional  
DCC - Universidade Federal de Minas Gerais

matheusaquino199@gmail.com.br 2018046513

**Abstract.** *In this report we will briefly discuss the implementation of the assignment 1 of the subject Information Retrieval. The problem consisted into implementing a simple crawler algorithm to collect URLs and its respective titles using the Chilkat library.*

**Resumo.** *Nesse relatório iremos discutir brevemente a implementação do Trabalho Prático 1 da disciplina Recuperação da Informação. O problema consistia em desenvolver um simples coletor de páginas web para encontrar URLs e seus respectivos títulos utilizando a biblioteca Chilkat.*

## 1. Introdução

O problema proposto no Trabalho Prático 1 consistia no desenvolvimento de um simples *Crawler*, que deveria ser implementado a partir do uso da biblioteca *Chilkat*, onde seriam dados dois parâmetros de entrada: uma *URL*  $s$  e um número inteiro  $n$ .

Assim, deveríamos desenvolver um algoritmo que a partir da página inicial  $s$  imprimisse  $n + 1$  *URLs* de enlaces, *links*, presentes na página incluindo o enlace inicial passado como entrada e seus respectivos títulos, e ao final imprimisse o tempo de *Crawling* médio  $\mu T$  gasto para pesquisar cada um dos enlaces encontrados.

## 2. Implementação

De um modo geral a implementação do algoritmo consistiu basicamente em um *loop* de repetição que para cada iteração era realizado o *Crawling* do próximo potencial *link URL*.

Desse modo, seja  $k$  o número de páginas coletadas com sucesso e  $t_i$  o tempo gasto para realizar o *Crawling* da  $i$ -ésima página, realizamos o seguinte: Caso seja encontrado um novo enlace  $l_i$  com sucesso, imprimimos o seu título e sua respectiva *URL*, e além disso, incrementamos o número de páginas pesquisadas com êxito e armazenamos o tempo de execução do *Crawling*. Caso contrário, imprimimos uma mensagem de erro relativa ao evento ocorrido, i.e, caso não haja mais páginas a serem pesquisadas ou um erro foi detectado.

Destarte, ao final da execução do *loop* de coleta principal, imprimimos o tempo médio em segundos  $\mu T$  de *Crawling* para cada enlace pesquisado com sucesso.

$$\mu T = \frac{1}{k} \sum_{i=0}^{k-1} t_i$$

Valor esse que é ajustado por um fator de  $10^{-6}$ , haja vista que a função utilizada da biblioteca *chrono* para computar o tempo, considera a execução em microssegundos.

### 3. Resultados

Foram realizados múltiplos testes para ilustrar o funcionamento do algoritmo em diferentes cenários.

Primeiro cenário, URL inicial "*www.bbc.com*" e  $n = 20$ ,

Successfully crawled links: 21  
Total crawling execution time(s): 29.1106  
Average crawling execution time(s): 1.38622

Segundo cenário, URL inicial "*www.goodreads.com*" e  $n = 20$ ,

Successfully crawled links: 21  
Total crawling execution time(s): 54.0347  
Average crawling execution time(s): 2.57308

Terceiro cenário, URL inicial "*ufmg.br*" e  $n = 20$ ,

Successfully crawled links: 21  
Total crawling execution time(s): 8.4789  
Average crawling execution time(s): 0.403757

A diferença no tempo de execução do *Crawling* para os diferentes cenários pode ser explicada por características particulares dos três sítios. O primeiro e segundo sítio possuem hospedagem em domínios não brasileiros, o que leva a uma latência e tempo de requisição superior ao do terceiro que possui uma hospedagem local.

Além disso, é notável que o primeiro sítio possui abrangência e número de acessos superior ao segundo, haja vista que é um jornal mundialmente conhecido, enquanto o segundo consiste em uma rede social de nicho. Assim, o primeiro sítio potencialmente possui um servidor mais adaptado e otimizado para o acesso dos seus usuários.

### 4. Conclusões

A partir desse trabalho foi possível ter um primeiro contato com coletores *Web*, que irão dar base para atividades futuras, e que por sua vez são parte fundamental de máquinas de busca modernas. Ademais, foi interessante entender e analisar resultados para diferentes sítios em cenários diversos, para perceber a complexidade e o comportamento de *Crawlers* na *Web* em variados contextos.

## 5. Código fonte implementado em C++

---

```
.
#include <CkSpider.h>
#include <iostream>
#include <chrono>
#include <string>

using namespace std;

void pageDisplay(int idx, string url, string title){ //Display Title and URL
    cout << "Page " << idx << "\n";
    cout << "Title: " << title << "\r\n";
    cout << "URL: " << url << "\r\n" << "\r\n";
}

int main(){
    string firstURL; cin >> firstURL; //Read URL
    int n; cin >> n; //Read number of additional pages to be crawled

    CkSpider spider; //Create spider and add the first URL to be collected
    spider.Initialize(firstURL.c_str());
    spider.AddUnspidered(firstURL.c_str());

    //Initialize execution time counter and successful crawled pages counter
    double exeTime = 0, k = 0;

    for (int i = 0; i < n+1; i++){ //Crawl through n+1 url link webpages
        //set initial crawling time and final crawling time
        auto initialExeTime = chrono::high_resolution_clock::now();
        bool found = spider.CrawlNext();
        auto finalExeTime = chrono::high_resolution_clock::now();

        //If a link page is successfully found, we display it,
        if (found){ //and add the crawling time to our execution time counter
            pageDisplay(i, spider.lastUrl(), spider.lastHtmlTitle());
            exeTime += chrono::duration_cast<chrono::microseconds>(finalExeTime -
                initialExeTime).count();
            k++;
        }
        else { //There're no more page links to be crawled in our initial url
            if (!spider.get_NumUnspidered()){
                cout << "There're no more pages to be crawled" << endl;
                break;
            }
            else //Or an error has been found
                cout << spider.lastErrorText() << "\r\n" << endl;
        }

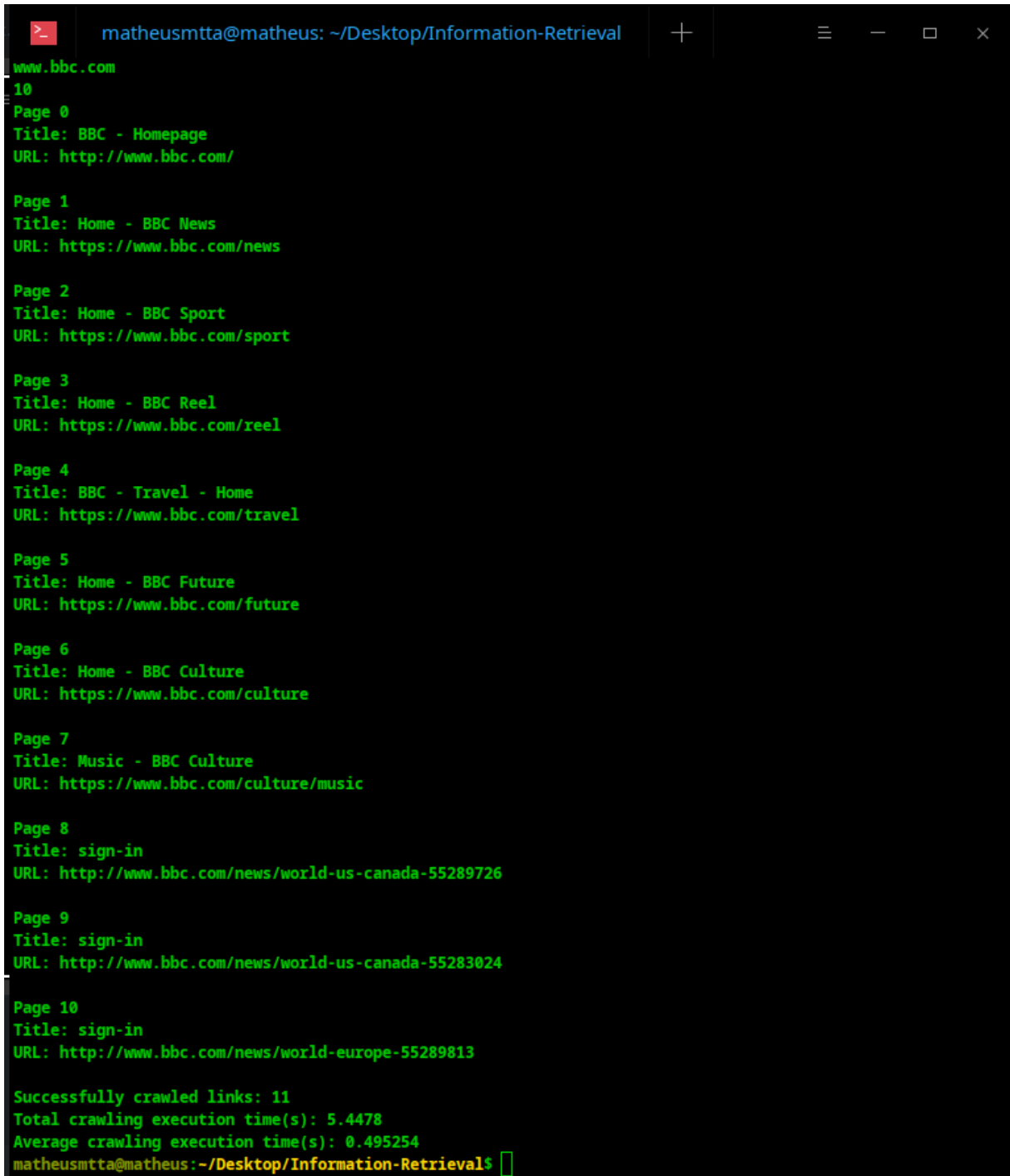
        spider.SleepMs(1000);
    }
    //Find total and average crawling execution time
    double avgExeTime = (exeTime*1e-6)/k;

    cout << "Successfully crawled links: " << k << endl;
    cout << "Total crawling execution time(s): " << exeTime*1e-6 << endl;
    cout << "Average crawling execution time(s): " << avgExeTime << endl;

    return 0;
}
```

---

## 6. Apêndice



```
matheusmtta@matheus: ~/Desktop/Information-Retrieval
www.bbc.com
10
Page 0
Title: BBC - Homepage
URL: http://www.bbc.com/

Page 1
Title: Home - BBC News
URL: https://www.bbc.com/news

Page 2
Title: Home - BBC Sport
URL: https://www.bbc.com/sport

Page 3
Title: Home - BBC Reel
URL: https://www.bbc.com/reel

Page 4
Title: BBC - Travel - Home
URL: https://www.bbc.com/travel

Page 5
Title: Home - BBC Future
URL: https://www.bbc.com/future

Page 6
Title: Home - BBC Culture
URL: https://www.bbc.com/culture

Page 7
Title: Music - BBC Culture
URL: https://www.bbc.com/culture/music

Page 8
Title: sign-in
URL: http://www.bbc.com/news/world-us-canada-55289726

Page 9
Title: sign-in
URL: http://www.bbc.com/news/world-us-canada-55283024

Page 10
Title: sign-in
URL: http://www.bbc.com/news/world-europe-55289813

Successfully crawled links: 11
Total crawling execution time(s): 5.4478
Average crawling execution time(s): 0.495254
matheusmtta@matheus:~/Desktop/Information-Retrieval$
```

Figure 1. Saída no Terminal dada pela entrada URL = "www.bbc.com", n = 10.

```
matheusmtta@matheus: ~/Desktop/Information-Retrieval
www.goodreads.com
10
Page 0
Title: Goodreads | Meet your next favorite book
URL: http://www.goodreads.com/

Page 1
Title: Forgot Password
URL: http://www.goodreads.com/user/forgot_password

Page 2
Title: Terms of Use
URL: http://www.goodreads.com/about/terms

Page 3
Title: Best Books 2020 ♦ Goodreads Choice Awards
URL: http://www.goodreads.com/choiceawards/best-books-2020?int=gca_signed_out_hp

Page 4
Title: Best Fiction 2020 ♦ Goodreads Choice Awards
URL: http://www.goodreads.com/choiceawards/best-fiction-books-2020?int=gca_signed_out_hp

Page 5
Title: Best Romance 2020 ♦ Goodreads Choice Awards
URL: http://www.goodreads.com/choiceawards/best-romance-books-2020?int=gca_signed_out_hp

Page 6
Title: Best Science Fiction 2020 ♦ Goodreads Choice Awards
URL: http://www.goodreads.com/choiceawards/best-science-fiction-books-2020?int=gca_signed_out_hp

Page 7
Title: Best Mystery & Thriller 2020 ♦ Goodreads Choice Awards
URL: http://www.goodreads.com/choiceawards/best-mystery-thriller-books-2020?int=gca_signed_out_hp

Page 8
Title: Best Nonfiction 2020 ♦ Goodreads Choice Awards
URL: http://www.goodreads.com/choiceawards/best-nonfiction-books-2020?int=gca_signed_out_hp

Page 9
Title: Best Humor 2020 ♦ Goodreads Choice Awards
URL: http://www.goodreads.com/choiceawards/best-humor-books-2020?int=gca_signed_out_hp

Page 10
Title: Best Fantasy 2020 ♦ Goodreads Choice Awards
URL: http://www.goodreads.com/choiceawards/best-fantasy-books-2020?int=gca_signed_out_hp

Successfully crawled links: 11
Total crawling execution time(s): 30.0065
Average crawling execution time(s): 2.72787
matheusmtta@matheus: ~/Desktop/Information-Retrieval$
```

Figure 2. Saída no Terminal dada pela entrada URL = "www.goodreads.com", n = 10.

```
matheusmtta@matheus: ~/Desktop/Information-Retrieval
ufmg.br
10
Page 0
Title: UFMG - Universidade Federal de Minas Gerais
URL: http://ufmg.br/

Page 1
Title: UFMG - Universidade Federal de Minas Gerais - International visitors
URL: http://ufmg.br/international-visitors

Page 2
Title: UFMG - Universidade Federal de Minas Gerais - Formas de Ingresso
URL: http://ufmg.br/cursos/formas-de-ingresso

Page 3
Title: UFMG - Universidade Federal de Minas Gerais - Cursos
URL: http://ufmg.br/cursos

Page 4
Title: UFMG - Universidade Federal de Minas Gerais - Vida Acadêmica
URL: http://ufmg.br/vida-academica

Page 5
Title: UFMG - Universidade Federal de Minas Gerais - Pesquisa e Inovação
URL: http://ufmg.br/pesquisa-e-inovacao

Page 6
Title: UFMG - Universidade Federal de Minas Gerais - Extensão
URL: http://ufmg.br/extensao

Page 7
Title: UFMG - Universidade Federal de Minas Gerais - Cultura
URL: http://ufmg.br/cultura

Page 8
Title: UFMG - Universidade Federal de Minas Gerais - Coronavírus
URL: https://ufmg.br/coronavirus

Page 9
Title: UFMG - Universidade Federal de Minas Gerais - Em sua primeira edição virtual, UFMG Jovem premia 36 trabalhos
URL: http://ufmg.br/comunicacao/noticias/em-sua-primeira-edicao-virtual-ufmg-jovem-premia-36-trabalhos

Page 10
Title: UFMG - Universidade Federal de Minas Gerais - Programa da Fafich ganha prêmio por iniciativas que promovem acesso à justiça
URL: http://ufmg.br/comunicacao/noticias/programa-da-fafich-ganha-premio-para-iniciativas-de-promocao-do-acesso-a-justica

Successfully crawled links: 11
Total crawling execution time(s): 4.37366
Average crawling execution time(s): 0.397606
matheusmtta@matheus:~/Desktop/Information-Retrieval$
```

Figure 3. Saída no Terminal dada pela entrada URL = "ufmg.br", n = 10.