Universitatea “Alexandru Ioan Cuza”, Iaşi

Facultatea de informatică

Surf

Parcurgere si procesare distribuita a paginilor web pentru extragerea informatiilor

Ovidiu Pricop

29 August 2016

Declaraţie privind originalitate şi respectarea drepturilor de autor

Prin prezenta declar că Lucrarea de licenţă cu titlul "Surf" este scrisă de mine şi nu a mai fost prezentată niciodată la o altă facultate sau instituţie de învăţământ superior din ţară sau străinătate. De asemenea, declar că toate sursele utilizate, inclusiv cele preluate de pe Internet, sunt indicate în lucrare, cu respectarea regulilor de evitare a plagiatului:

- toate fragmentele de text reproduse exact, chiar şi în traducere proprie din altă limbă, sunt scrise între ghilimele şi deţin referinţa precisă a sursei;
- reformularea în cuvinte proprii a textelor scrise de către alţi autori deţine referinţa precisă;
- codul sursă, imaginile etc. preluate din proiecte open-source sau alte surse sunt utilizate cu respectarea drepturilor de autor şi deţin referinţe precise;
- rezumarea ideilor altor autori precizează referinţa precisă la textul original.

Iaşi, 29 August 2016

Ovidiu Pricop

Declaraţie de consimţământ

Prin prezenta declar că sunt de acord ca Lucrarea de licenţă cu titlul "Surf", codul sursă al programelor şi celelalte conţinuturi (grafice, multimedia, date de test etc.) care însoţesc această lucrare să fie utilizate în cadrul Facultăţii de Informatică. De asemenea, sunt de acord ca Facultatea de Informatică de la Universitatea Alexandru Ioan Cuza Iaşi să utilizeze, modifice, reproducă şi să distribuie în scopuri necomerciale programele-calculator, format executabil şi sursă, realizate de mine în cadrul prezentei lucrări de licenţă.

Iaşi, 29 August 2016

Ovidiu Pricop

Rezumat

La nivelul internetului, traficul global a crescut de aproximativ 850 de ori in perioada 2000 - 2015 [1]. World wide web-ul reprezinta o parte semnificativa a volumului de informatii interschimbate pe internet. In anul 2015 existau peste jumatate de miliard de situri web accesibile [2]. Fiecare pagina web raspunde anumitor nevoi (sociale, financiare, educationale etc.). O singura sursa de informatii este uneori suficienta pentru a raspunde nevoilor unui utilizator. Alteori, este necesar un ansamblu de surse informative (e.g. newsletter-e zilnice din 5 surse diferite), pentru a urmari un subiect din mai multe perspective sau a-i intregi continutul.

Prezenta lucrare urmareste elaborarea unui serviciu web distribuit in cloud pentru parcurgerea, selectarea, colectarea si indexarea informatiilor la nivelul world wide web-ului.

Contents

Rezumat **3**

Introducere **5**
1 Crawling-ul paginilor web 6
1.1 Puncte de pornire 6
1.2 Parcurgere 6
1.3 Selectia informatiilor 7
1.4 Viteza, politici de respingere si drepturi de autor 7
1.5 Paralelizare 8
1.6 Logarea actiunilor si retentia datelor 8
2 Crawling in Amazon Web Services 10

Introducere

Un serviciu web reprezinta o componenta functionala ce indeplineste anumite sarcini. Comunicarea cu un serviciu web se realizeaza independent de platforma, limbajul de programare sau sistemul de operare pe care este dezvoltat. Schimbul de informatii se realizeaza prin mesaje text ce respecta un format standardizat precum xml[1] sau json[2].

Aplicatia "Surf" reprezinta un serviciu web specializat in web crawling[3], dezvoltat folosind tehnologii cloud din cadrul Amazon Web Services. Se urmareste crearea unui serviciu web cu disponibilitate permanenta, scalabil si de inalta putere computationala care sa orchestreze colectarea distribuita de informatii din aria definita de utilizator.

Comunicarea cu aplicatia "Surf" se realizeaza prin intermediul unui API Restful[4] construit pe platforma AWS[5] API Gatway. Autentificarea utilizatorilor se va realiza printr-un serviciu OpenID Connect (e.g. Google, Facebook etc.). Autorizarea va avea ca principala componenta AWS IAM. Utilizatorilor le vor fi repartizate, in functie de privilegiile asociate cu cheia de autentificare, o serie de roluri (i.e. drepturi de access asupra resurselor din cadrul serviciului "Surf"). Executia codului propriu-zis, gazduit de functii AWS Lambda, va interactiona cu baza de date no-sql AWS DynamoDB pentru a eficientiza accesul la informatii cheie (metadate crawling, date autentificare etc.). Mediul de procesare distribuita va fi sustinut de AWS Simple Workflow Service si configurat dupa preferintele utilizatorului. Datele extrase din procesul de web-crawling vor fi salvate in mediul persistent de stocare AWS S3. Evenimentele legate de parcurgerea siturilor vor fi expuse, ca metadate, intr-o coada AWS SQS si vor fi accesibile utilizatorilor prin procesul de long-polling asupra acestei cozi.

[1] Extensible Markup Language - https://www.w3.org/XML/
[2] JavaScript Object Notation - http://www.json.org/
[3] https://en.wikipedia.org/wiki/Web_crawler
[4] https://en.wikipedia.org/wiki/Representational_state_transfer
[5] Amazon Web Services

1 Crawling-ul paginilor web

Ca sens general, un crawler web reprezinta un program care parcurge, prin cereri succesive, situri web. Programatic, vom considera urmatoarele aspecte cheie in proiectarea unui crawler distribuit:

1. Punctele de pornire in parcurgerea siturilor web;
2. Adancimea maxima a parcurgerii recursive a siturilor (i.e. cea mai indepartata pagina la care se poate ajunge, de la punctul de pornire, prin accesarea succesiva a legaturilor de tipul hyperlink;
3. Politica de selectie a informatiilor din paginile parcurse[3];
4. Viteza de crawling, fisierul *robots.txt*[6] si respectarea drepturilor de autor;
5. Politica de paralelizare a procesului de web crawling;
6. Politica de retentie temporara a rezultatelor parcurgerii siturilor web si logare a actiunilor serviciului de crawling;

In cele ce urmeaza, se va descrie particularizarea aspectelor generale enumerate mai sus in contextul serviciului web "Surf". Se va crea, astfel, contextul dezvoltarii aplicatiei si se vor puncta principalele componente functionale implicate.

1.1 Puncte de pornire

Pentru a putea parcurge siturile web, crawler-ul "Surf" necesita unul sau mai multe puncte de pornire. Un punct de pornire este definit printr-un URL[7] si este furnizat, ca input din partea utilizatorului, la initializarea crawler-ului.

1.2 Parcurgere

Crawler-ul web "Surf" are ca scop extragerea informatiilor cerute de catre utilizator. Intrucat utilizatorul are posibilitatea de a furniza, ca punct de pornire, un domeniu web relevant pentru informatia cautata, parcurgerea recursiva se va executa in maniera breadth-first. Asadar, crawler-ul va vizita toate legaturile de tip hyperlink din pagina curenta a parcurgerii inainte de a accesa legaturile din pagina urmatoare din punct de vedere ierarhic. Adancimea maxima a arborelui de legaturi realizat prin parcurgerea URL-urilor va fi definita de catre utilizator, la initializarea sesiunii de crawling.

[6] http://www.robotstxt.org/robotstxt.html
[7] https://en.wikipedia.org/wiki/Uniform_Resource_Locator

1.3 Selectia informatiilor

Selectia informatiilor necesita parcurgerea siturilor web aflate la adresele URL pe care crawler-ul le are in vedere, parsarea informatiilor in functie de tipul lor (e.g. html, xml, json, text) si extragerea blocurilor de continut aferente.

Un "bloc de continut" asociat unui URL reprezinta o parte a intregului continut aflat la URL-ul respectiv, filtrata dupa anumite caracteristici stabilite de utilizatorul serviciului web. Aceste caracteristici pot fi:

- Pentru fisiere HTML/XML:
  - selectori CSS/jQuery
  - cuvinte cheie
  - o combinatie logica a punctelor de mai sus (e.g. toate tagurile <p> in care se afla cuvintele cheie "om" si "luna")
- Pentru fisiere text:
  - cuvinte cheie

Pentru tipurile de continut enumerate mai sus sau pentru alte tipuri de continut aflate la adresele URL vizitate de crawler, utilizatorul poate defini filtre bazate pe continutul textual al URL-ului. Spre exemplu, se pot evita toate URL-urile care nu satisfac o anumita expresie regulata sau un anumit tip de continut MIME[8] (e.g. "image/png").

Evitarea selectarii informatiilor duplicate se amelioreaza prin normalizarea[9] URL-urilor parcurse de catre "crawler".

1.4 Viteza, politici de respingere si drepturi de autor

Siturile web pot implementa variate modalitati de contracarare a incercarilor de crawling. Aplicatia "Surf" incearca sa minimizeze riscul de respingere a cererilor de accesare a anumitor resurse web printr-o implementare neintruziva a procesului de crawling. Cateva aspecte esentiale care sunt luate in considerare in ceea ce priveste o astfel de implementare sunt urmatoarele:

- Minimizarea volumului de date preluat de pe un anumit domeniu prin diferite metode de filtrare a linkurilor urmarite (e.g. o anumita structura a URL-ului, un anumit tip de date care se gaseste la URL-ul respectiv);

[8] https://en.wikipedia.org/wiki/Media_type
[9] https://en.wikipedia.org/wiki/URL_normalization

- Introducerea pauzelor temporale aleatoare intre accesari succesive a datelor apartinand aceluiasi domeniu.

Crawler-ul web "Surf" poate satisface numeroase cerinte ale utilizatorilor. O parte dintre aceste cerinte poate veni din partea sistemelor anti-malware. In acest caz, nu se doreste respectarea fisierului *robots.txt* (utilizat drept referinta, pentru crawleri, asupra URL-urilor accesibile ale domeniului vizitat), deoarece exita pericolul ca un site malitios sa blocheze o eventuala scanare. De aceea, aplicatia "Surf" va putea fi configurata in ceea ce priveste ignorarea fisierului *robots.txt* in procesul de parcurgere a unui domeniu.

Crawler-ul "Surf" implementeaza un mecanism de *blacklisting*[10]. Siturile web sau domeniile care interzic procesul de crawling (e.g. prin ToS[11]) vor fi adaugate unei liste de excluziune din procesul de parcurgere executat de crawler.

1.5 Paralelizare

"Surf" implementeaza un mecanism de crawling distribuit. Mai exact, exista posibilitatea de a imparti sarcinile de parcurgere a paginilor web prin rularea concurenta a mai multor instante de functii Lambda[12].

Exista doua moduri bine cunoscute de alocare a sarcinilor de web crawling: statica si dinamica[4]. Aplicatia "Surf" implementeaza un mecanism de distribuire statica a sarcinilor:

> Fie n, *gradul de paralelizare*[13] configurat pentru executia curenta a web crawler-ului. Alocarea statica va asocia fiecarui URL candidat pentru crawling un numar intreg in intervalul $[1,n]$ reprezentand identificatorul uneia dintre cele n instante paralele ale web crawler-ului. Sarcina va fi distribuita pe acea instanta si va fi executata.

1.6 Logarea actiunilor si retentia datelor

In timpul executiei, crawler-ul genereaza evenimente ce sunt adaugate la istoricul executiei curente. Aceste evenimente sunt utile, in primul rand, in cazul in care executia instantei curente a crawler-ului esueaza. Sarcinile de crawling web pot dura o perioada considerabila si nu este dezirabila refacerea intregului

[10] https://en.wikipedia.org/wiki/Blacklisting
[11] https://en.wikipedia.org/wiki/Terms_of_service
[12] Amazon Web Services Lambda Functions
[13] Numarul maxim de executii simultane de sarcini de crawling

proces de parcurgere a siturilor, atat timp cat exista rezultate partiale; executia se poate relua pornind de la ultimul eveniment valid inregistrat in istoric. In al doilea rand, istoricul este util pentru utilizator, deoarece il poate avertiza in legatura cu statutul executiei curente a web crawler-ului.

Datele rezultate in urma procesului de crawling al paginilor web sunt salvate in serviciul persistent de stocare S3, din cadrul Amazon Web Services. Deoarece aceste date pot deveni volumnioase (in functie de politica de selectie configurata), aplicatia "Surf" salveaza metadate asociate acestor rezultate intr-o tabela DynamoDB. Simultan, o parte dintre metadate sunt plasate intr-o coada SQS[14]. Utilizatorul poate sa efectueze long-polling asupra acestei cozi pentru a extrage informatiile legate de procesul de crawling. Pe baza acestor informatii (metadate), utilizatorul are optiunea de a accesa rezultatele complete corespunzatoare rularii crawler-ului web, aflate in S3.

[14]Amazon Simple Queue Service

2 Crawling in Amazon Web Services

Bibliografie

[1] "Visual Networking Index", Cisco Systems

[2] http://www.internetlivestats.com/total-number-of-websites/

[3] https://en.wikipedia.org/wiki/Web_crawler#Selection_policy

[4] https://en.wikipedia.org/wiki/Distributed_web_crawling