Sisteme de regăsire a informațiilor
Colectarea datelor Web

Duliba Ioan-Adrian

Rezumat

Titlul temei şi scopul aplicației

Tema propusă este realizarea unei aplicații ce transferă informațiile din paginile web în baza de date cu scopul de a le indexa după conținut, obiectivul fiind de a salva cât mai rapid şi eficient datele paginilor accesate.

Scopul acestei lucrări este de a studia principalele compunente ale unui robot Web şi de a propune soluții de implementare pentru toate aceste componente.

Descrierea aplicației

Activitatea unui robot Web este cea de a colecta, într-un mod automat, resurse web. Un fir de execuție ce foloseste un crawler începe prin a lua o adresă URL de la frontieră şi de a prelua pagina web la acea adresă, utilizând în general protocolul HTTP. Apoi analizează textul paginii si sunt extrase link-urile, după care sunt trecute printr-o serie de teste pentru a se decide dacă trebuie adăugate la frontiera URL. Atunci când o adresă este adăugată la cozile de explorare, îi este atribuită o prioritate în baza căreia este eventual eliminată din frontieră pentru prelucrare.

Tehnologii folosite şi structura proiectului

Pentru modulul aplicației ce are rol de client DNS , ca limbaj de programare am ales limbajul C, datorită eficienței codului obiect pe care îl poate genera (redus ca dimensiune şi rapid in executie) şi pentru portabilitatea sa. Un alt avantaj al acestuia este faptul că permite programarea la un nivel mai scăzut decât alte limbaje, mai apropiat de hardware: operații pe biți, acces direct la memorie.

Pentru celelalte module ale aplicației, cache-ul DNS, cel de aducere şi parsare a paginilor WEB, cele de filtrare a link-urilor şi pentru frontiera URL am ales limbajul Java.

Un cache DNS este o bază de date temporară ce conține un index al tuturor site-urilor vizitate şi al IP-urilor corespunzătoare acestora. Comunicarea informațiilor dintre clientul DNS şi cache-ul respectiv se realizează printr-un obiect de tip JSON , transferat printr-un fişier, apoi acestea ajung să fie stocate într-o structură de date de tip arbore B+, deoarece operațiile de inserare şi ştergere sunt relativ eficiente in toate condițiile. Protocolul HTTP este folosit în mod obişnuit pentru sistemele informatice distribuite unde performanța poate fi îmbunătățită prin reutilizarea răspunsurilor din cache. Pentru a decide dacă răspunsul este valid , trebuie să comparăm durata de viață a prospețimii sale („*max-age*”) cu vârsta sa. Informațiile sunt păstrate într-un fişier, fiind serializate pentru a optimiza lucrul cu acestea. Deoarece regulile impuse de robots.txt se schimbă rar, am implementat un modul ce memorează

informațiile privind acest aspect în cache timp de până la o zi, însă acestea pot fi stocate pentru mai mult timp în situațiile în care nu este posibilă actualizarea versiunii (de exemplu, datorită erorilor de ”*timeout*” sau erorilor *5xx*). Pentru a realiza acestu lucru, modulul foloseşte o bază de date MongoDB.

Toate aceste module realizate cu ajutorul limbajului Java sunt proiecte de tip Maven, avantajele principale fiind descrierea procesului de *build* a *softwareului* şi a dependenţelor acestuia, descarcând dinamic *bibliotecile* si *plug-in-uri* necesare, din unul sau mai multe *repository-uri*.

Ca şi mediu de dezvoltare integrat am folosit Eclipse, deoarece este o platformă deschisă, un mediu integrat de dezvoltare (*I.D.E.*) ce oferă facilități de management al spațiului de lucru, creare, lansare şi depanare aplicații.

Cu alte cuvinte aplicația foloseşte limbaje independente de platforma de lucru, având capacitatea să ruleze, fără nici o modificare, pe sisteme diferite cum ar fi Windows, UNIX sau Macintosh.