Blaž Štempelj in Denis Grabljevec

Seminarska naloga: Implementacija spletnega pajka

V seminarski nalogi je predstavljen spletni pajek, ki se “sprehaja” po povezavah spletnih strani z domeno gov.si. Ko pajek obišče spletno stran, iz nje pridobi vse povezave, slike in dokumente. Za pridobivanje povezav pajek uporablja strategijo znano pod imenom iskanje v širino, rezultate iskanja pa shranjuje v PostgreSQL relacijsko podatkovno bazo, ki se nahaja na Amazonovem oblaku AWS.

Uvod

Spletni pajki so programi, ki obiskujejo povezave na spletnih straneh. Ko obiščejo neko spletno stran, si zapomnijo vse povezave na podstrani, da jih kasneje lahko obiščejo. To ponavljajo dokler jim ne zmanjka povezav. Poznamo več vrst spletnih pajkov. Najbolj osnovna delitev je glede na vsebino. Univerzalni spletni pajki obiščejo katerokoli stran, saj jih sama vsebina ne zanima. Zanimajo jih čisto vse povezave. Fokusirani in področni spletni pajki pa preiskujejo le spletne strani, ki se osredotočajo na določeno tematiko (npr. novice).

Arhitektura

Spletni pajek, ki smo ga izdelali, vsebuje naslednje večje komponente:

- frontier
- bralnik kazala spletne strani
- bralnik datoteke robots.txt
- relacijska podatkovna baza SQL
- večnitenje

Frontier je podatkovna struktura, ki hrani povezave, katere bo spletni pajek obiskal v bližnji prihodnosti. Poleg še ne obiskanih povezav, frontier hrani tudi že obiskane povezave v normalizirani obliki.

Glavna naloga bralnika kazala spletne strane je, da iz kazala izlušči vse povezave do zanimivih podstrani. Te povezave nato doda v frontier, s čimer zagotovi, da spletni pajek najprej obišče te podstrani in se šele nato loti ostalih manj pomembnih podstrani.

Drugi bralnik, ki ga uporabljamo je bralnik datoteke robots.txt. Ta datoteka nam pove do katerih podstrani imamo dostop. Poleg specificira koliko časa mora pajek počakati preden lahko obišče naslednjo povezavo in s tem prepreči preobremenitev spletnega strežnika.

Za shranjevanju vseh povezav, slik in ostalih dokumentnih datotek, ki jih je pajek našel med preiskovanjem, sva uporabila relacijsko podatkovno bazo SQL.

Implementirani spletni pajek uporablja vzporedno arhitekturo s pomočjo nitenja, s čimer se število povezav, ki jih lahko preišče, drastično poveča.

Implementacija

Za implementacijo spletnega pajka smo se odločili uporabiti programski jezik Python [4], ker ponuja ogromno knjižnic, ki so nam olajšale in pohitrile implementacijo. Ker večina današnjih spletnih strani za popolno delovanje potrebuje JavaScript, smo uporabili orodje Selenium [1], ki omogoča izvajanje JavaScripta.

Najprej smo implementirali frontier s pomočjo strukture *deque*, ki je Pythonovska implementacija FIFO vrste, katera je glavni pogoj za implementacijo pajka, ki išče v širino. Seznam že obiskanih povezav je slovar tipa ključ-vrednost, kjer ključ predstavlja rezultat zgoščevalne funkcije SHA-1, katere vhod je normalizirana povezava.

Začetne povezave, ki se dodajo v frontier so:

- http://evem.gov.si,
- http://e-uprava.gov.si,
- http://podatki.gov.si,
- http://e-prostor.gov.si
- http://ursm.gov.si
- http://gsv.gov.si
- http://fu.gov.si
- http://mizs.gov.si
- http://mop.gov.si
- http://mzi.gov.si

Pri vseh ostalih podstraneh na katere naleti spletni pajek preverimo, ali so podstrani teh začetnih povezav, saj nas ostale povezave, ki vodijo na preostali del spleta ne zanimajo. Vsaka povezava, ki se dodan v frontier gre najprej čez postopek normalizacije, kjer:

- odstranimo predpono *www*,
- *https* spremenimo v *http* (ne uporabljamo certifikatov),
- prekodiramo nestandardne znake,
- odstranimo pripono vrat do storitve in osnovno stran index.html,
- in na koncu še dodamo poševnico /, če gre za direktorij.

Za branje datoteke robots.txt uporabljamo Pythonovsko knjižnico RobotFileParser [5], ki omogoča preverjanje ali lahko naš pajek dostopa do neke povezave in definicijo zamika naslednjega dostopa do nove povezave. Iz datoteke robots.txt bralnik kazala spletne strani prebere vse povezave do vseh obstoječih kazal in njihove povezave doda v frontier.

Arhitektura pajka je zasnovana večnitno, s čimer omogoča, da se hitreje premika po povezavah. Število niti, ki jih želimo uporabiti, specificiramo kot argument programu. Vse niti uporabljajo isti frontier. S tem smo se izognili potrebi po sinhronizaciji frontierjev.

Podatkovna baza PostgreSQL [3] se nahaja na Amazonovem oblaku AWS [2]. V tabelo *page_type* je bil dodan tip strani, in sicer *IMAGE.* Tako je bolj jasno, katere stran se povezujejo s tabelo *page_data* in katere s tabelo *image*.

Rezultati

Domena	**Število povezav**	**HTML**	**Slika**	**PDF**	**DOC**	**PPT**
e-prostor.gov.si	618	207	57	228	34	1
mzi.gov.si	13770	5166	5415	1435	231	7
mop.gov.si	18526	3914	8710	4990	575	9
mizs.gov.si	29346	7568	11678	6458	2286	131
fu.gov.si	3330	839	1158	571	516	0
evem.gov.si	6511	5757	113	79	110	0
e-uprava.gov.si	1878	1878	0	0	0	0
podatki.gov.si	19463	19172	9	22	7	0
gsv.gov.si	414	110	152	51	85	0
ursm.gov.si	1256	536	82	449	125	5

Tabela 1: Rezulati preiskovanja.

Rezultati so pokazali, da se največ povezav nahaja na domeni *mizs.gov.si*, vendar večina teh povezav ne predstavlja spletnih strani, temveč so to slike ali dokumentni formati. Druga stran z največ povezavami je *podatki.gov.si*, kjer pa je le malo slik in dokumentov, večino namreč povezav predstavljajo podstrani. Najmanj povezav vsebuje domena *gsv.gov.si*, kjer večino teh povezav predstavlajo slike ali dokumenti. Le tretjina povezav so podstrani. Zanimivo je tudi, da na domeni *e-uprava.gov.si*, pajek ni naletel na ne na sliko, ne na dokument. Kar pa ni rečeno, da domena ne vsebuje slik ali dokumentov. Najverjetneje slike niso shranjene v html značkah img po katerih pajek išče slike.

Zaključek

Tekom implementacije smo nateli kar na nekaj težav. Največ preglavic so predstavljale povezave https, saj naš spletni pajek ne podpira rokovanja z SSL certifikati. V splošnem smo z implementacijo in rezultati zadovoljni. Se pa hkrati tudi zavedamo, da je še veliko prostora za izboljšave. Npr. zaradi načina implementacije se določeni zahtevki do istih povezav zgodijo večkrat, s čimer se upočasni delovanje spletnega pajka.

Literatura

[1] https://docs.seleniumhq.org/
[2] https://aws.amazon.com/
[3] https://www.postgresql.org/
[4] https://docs.python.org/3/
[5] https://docs.python.org/3/library/urllib.robotparser.html?highlight=robotparser#module-urllib.robotparser