Seminarska naloga 1 - Spletni pajek

Andraž Jelenc, Vid Križnar, Sandi Režonja

March 24, 2020

1 Uvod

Cilj seminarske naloge je bila izdelava pajka za preiskovanje spletišč domene gov.si. Pajek deluje po principu iskanja v širino in implementira vzporedno delovanje več pajkov. Napisan je v jeziku Python, za podatkovno bazo smo izbrali PostgreSQL 9. Oba tečeta vsak v svojem Docker kontejnerju. Nastavitve pajka, kot so začetne strani in število vzporednih pajkov, nastavimo v datoteki docker-compose.yml. Vse skupaj pa nato zaženemo z uporabo orodja Docker Compose.

2 Implementacija

2.1 Frontier

Frontier smo implementirali v bazi s tabelo `crawldb.page`. Sestavljajo ga tiste strani, katerih `page_type` atribut je bil nastavljen na FRONTIER. Spoštovanje zakasnitve smo implementirali z razširitvijo baze s tabelo `crawldb.server`, v katero smo shranili IP naslov strežnika in čas zadnjega obiska. Tabelo `crawldb.site` smo razširili s poljem želenega premora med obiski. Iskanje naslednje strani za obdelavo je iz fontierja vrnilo stran z najmanjšim indeksom, ki je zadoščala zakasnitvenem pogoju.

Pri dodajanju, pajek preskoči strani, ki niso iz domene gov.si. Če strežnika še ne pozna, najprej tega doda v bazo. Če ne pozna domene, doda novo in pri tem prenese vsebino strani robots.txt, poveča zakasnitveni pogoj, če je to zahtevano in poskusi najdi sitemap datoteko. Če le-ta obstaja, shrani pot do datoteke v bazo, vse URL naslove, ki jih le-ta vsebuje pa doda v frontier. Nato nadaljuje z dodajanjem trenutne strani. Če domena premore sitamap, robots.txt prepoveduje obisk strani ali pa je stran že v frontierju, je ne doda. Če smo dodali novo stran, ustvarimo nov vnos za povezavo med novo stranjo in stranjo, kjer smo našli ta URL naslov.

2.2 Vzporednost

Delovanje več vzporednih pajkov smo zagotovili z večnitnostjo. Vsak pajek je deloval do izpolnitve zaustavitvenega pogoja – praznega frontierja. Vendar se zaradi več vzporednih pajkov lahko zgodi, da je frontier prazen le začasno. Zato pajek frontier večkrat preveri in se ustavi šele po določenem številu poskusov. Ko pajek iz frontierja prenese URL naslov naslednje strani, pri tem posodobi zadnji čas dostopa do strežnika in nastavi status strani na PROCESSING. Nato začne z obdelavo strani.

2.3 Obdelava

S selenium-wire (razširitev ogrodja Selenium) prenesemo vsebino strani, na katero kaže izbrani URL. Prenašanju velikih binarnih datotek se izognemo tako, da omejimo čas prenosa. Če je časovna omejitev presežena, strani nastavimo status `null`, s čim zabeležimo napako. Med nalaganjem sledimo preusmeritvam in vrnemo status in tip vsebine prvega odgovora, ki nas ne preusmerja drugam, če tak odgovor obstaja. Če status odgovora ni enak statusu 200 OK, v podatkovno bazo shranimo le tip vsebine in status, s samo vsebino pa se ne ukvarjamo. Sicer pa obravnavo ločimo na dva primera. Če gre za binarno datoteko nastavimo tip strani na BINARY in v primeru, da gre za PDF, DOC, DOCX, PPT ali PPTX datoteko v tabeli `page_data` dodamo zapis o najdenem dokumentu. Vsebine binarne datoteke ne shranjujemo. V primeru HTML strani pa preverimo, ali MD5 vrednost vsebine že poznamo (duplikat), nato pa iz vsebine izluščimo vse URL naslove iz atributov `href` in `onclick`. Pred dodajanjem najdenih

naslovov v frontier, relativne naslove pretvorimo v absolutne in nad naslovi izvedemo kanonizacijo, poleg tega naslovom odstranimo začetne `www`. Iz vsebine izluščimo tudi vse `src` atribute slik na strani. Za vsako sliko dodamo zapis v `crawldb.image`, kjer poleg naslova datoteke poskusimo iz njega izluščiti še tip slike.

3 Rezultati

Pajka smo pustili teči 82 ur, pri tem smo uporabili 4 niti. V tem času je preiskal povprečno 3 domene na uro in pri tem obdelal 886 strani na uro. Natančnejši podatki so na voljo v tabeli 1. Število vseh strani, ki jih zaradi različnih razlogov nismo uspeli prenesti v celoti (neodzivnost strežnika, časovni limit...) je navedeno v vrstici # z napako. Z orodjem Gephi smo vizualizirali povezanost med domenami. Na vizualizaciji 1 je prikazana povezanost domen, med katerimi je vsaj 50 linkov med spletnimi stranmi. Velikost imena domene predstavlja število strani, velikost povezave pa število linkov. Oboje je v logaritemski skali. Izvoz baze je zaradi velikosti naložen na Dropbox, na Github pa smo objavili le link.

	(vse domene)	gov.si	evem.gov.si	e-uprava.gov.si	e-prostor.gov.si
# domen	247	-	-	-	-
# strani / domeno	294	-	-	-	-
# vseh strani	72683	24369	6443	20939	516
# html strani	49105	20751	3937	17454	166
# duplikatov	372	132	35	0	10
# DOC	436	0	31	146	27
# DOCX	257	0	0	91	3
# PDF	1052	0	102	189	112
# PPT	1	0	0	0	1
# PPTX	1	0	0	0	0
# druge binarne	592	3	90	42	86
# slik	100229	28148	19535	17684	60
# slik / html stran	2,0	1,4	5,0	1,0	0,4
# z napako	20867	3483	2248	3017	111

Tab. 1: Statistični podatki pajkovega delovanja.

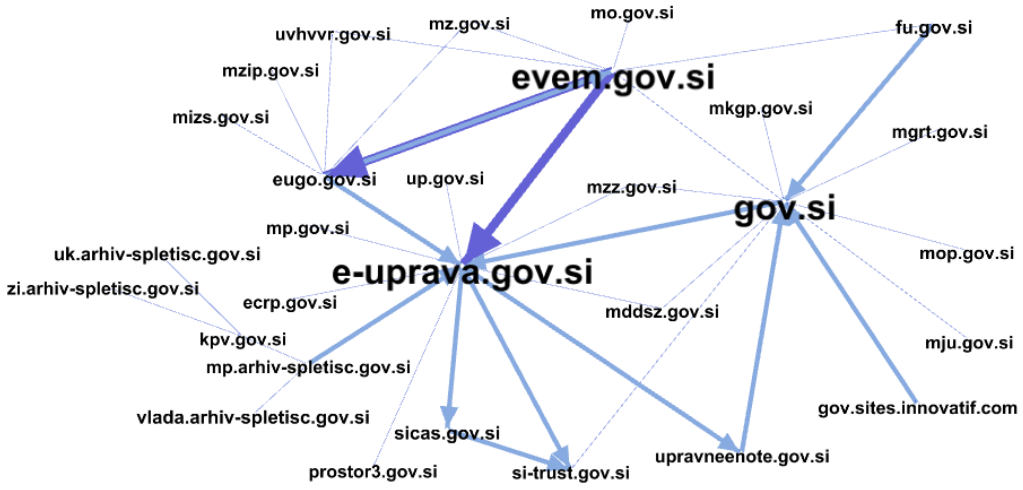

Fig. 1: Vizualizacija povezanosti domen.

4 Literatura

- https://gephi.org
- https://docs.python.org/3/library/threading.html
- https://pypi.org/project/psycopg2/
- https://pypi.org/project/selenium-wire/