Fakulteta za računalništvo in informatiko, Univerza v Ljubljani

Luka Dragar, Matej Vatovec in Aleksander Kovač

Spletni pajek

Iskanje in ekstrakcija podatkov s spleta: 1. domača naloga

MENTOR: prof. dr. Marko Bajec, doc. dr. Slavko Žitnik

1. Uvod

V poročilu je na kratko napisan povzetek izvedbe spletnega pajka za prvo domačo nalogo pri predmetu Iskanje in ekstrakcija podatkov s spleta.

2. Struktura pajka

Pajek je sestavljen iz treh delov, pri čemer strežnik odjemalcem dodeljuje spletne naslove, ki jih odjemalci prejmejo in iz naslova renderirajo spletno stran. Iz te strani odjemalci preberejo zahtevane podatke, ki jih potem strežnik shrani v podatkovno bazo Postgre SQL. Na strani se nahajajo pretežno novi spletni naslovi, ki jih odjemalci pošljejo strežniku, ta pa jih potem dodeljuje drugim (ali istemu odjemalcu), da pridobijo nove podatke.

2.1 Strežnik

Strežnik smo implementirali v programskem jeziku Python. Obsega povezavo s podatkovno bazo, kot tudi drugimi funkcionalnostmi, ki so naslednje:

- Dodeljevanje naslednje strani, ki še niso bile na vrsti za pridobitev vsebine
- Nastavljanje trenutnega statusa, kaj se s stranjo dogaja.
- Vstavljanje nove, še ne prebrane (sparsane?) strani.
- Posodabljaje informacij o straneh, ki so bile na novo o strani odkrite.
- Vstavljanje osnovnih podatkov o strani v podatkovno bazo.
- Preverjanje ali je trenutna stran ali druga vsebina podvojena.
- Vstavljanje slikovnega gradiva v bazo.
- Pridobivanje vsebine robots.txt.
- Izbris podatkov in ponastavitev osnovne sheme podatkovne baze.

Seveda so še na voljo podporne metode z dodatno logiko za obdelovanje podatkov. Strežnik lahko teče lokalno pri vsakem uporabniku in se povezuje na podatkovno bazo Postgre SQL gostovano v oblaku. Vse te funkcije uporabljajo odjemalci za prejemanje povezav za obdelavo in vstavljanje rezultatov v podatkovno bazo.

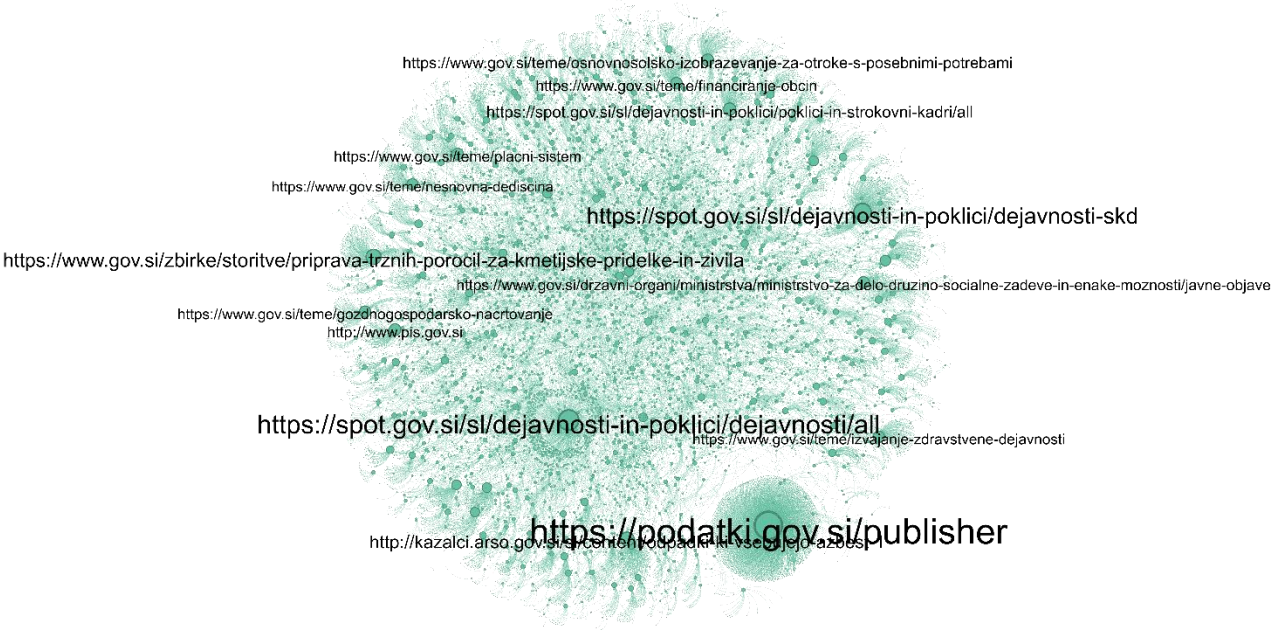
2.2 Odjemalci

Tako kot strežnik, so odjemalci napisani v programskem jeziku Python. Odjemalec od strežnika poizkuša od strežnika pridobiti URL, ki je na voljo in še ni bil raziskan, prav tako pa podatki iz njega še niso bili izluščeni. Pridobi ga na podlagi statusa naslova v bazi (status NOT_PARSED) in hkrati upošteva kdaj je bil naslov dodan v frontier (starejši naslovi imajo prednost – breadth first search). Prav tako upošteva, da v roku petih sekund ne vzame URL-ja iz istega IP naslova. Odjemalec generira spletno stran in iz nje izlušči podatke, ko so drugi vsebovani URL naslovi, slike (tudi kot base64 zapis), povezave do drugih dokumentov itd.) in jih vrne na strežnik, skupaj s HTML kodo in drugimi parametri, kot sta statusna koda (najpogosteje 200). Odjemalec potem od strežnika pridobi drug relevanten naslov in postopek se ponovi.

3. Statistika

	Pages	HTML	DUPLICATE	BINARY	PARSED	Images	PDF
Vse	35188	20611	152	14225	17484	47479	2984
gov.si	19163	7757	76	11316	9051	9144	2168
evem.gov.si	4414	4050	23	277	3088	11505	98
e-uprava.gov.si	1323	1295	0	15	900	912	7
e-prostor.gov.si	601	174	0	423	540	1258	186

4. Vizualizacija



Slika 1: Vizualizacija povezav spletnih strani

5. Težave

Med poganjanjem pajka in strežnika smo naleteli na naslednje težave:

- V podatkovno bazo so se nam prikradle tudi povezave, ki kažejo na spletne strani, kot so na primer facebook, twitter in druge. To je bila posledica tega, da nismo pravilno preverjali, kje se niz »gov.si« nahaja. Ne želenih povezav smo se v bodoče rešili s pomočjo regularnega izraza »^https?://(?:[a-z0-9](?:[a-z0-9-]{0,61}[a-z0-9])?\.)+gov\.si(?:$|/)«, ki je upošteval samo strani, ki prihajajo z domen https://*.gov.si.
- Izziv je bil tudi izgradnja mehanizma za sočasno dostopanje do baze različnih odjemalcev. Čeprav se tega nismo lotili z zaklepanjem podatkovne baze, kar bi pripeljejo do manjše verjetnosti sočasnega dostopa istih podatkov, smo problem rešili na enostavnejši način. Takoj, ko se iz frontierja vzame nov naslov za procesiranje, se status tega naslova spremeni in odraža, da se naslov trenutno obdeluje. Tako naslednji dostop do frontierja ne bo vrnil istega naslova. To ne drži le v trenutku, ko bi dva odjemalca dostopala do istega podatka v manjšem razmiku kot je možno spremeniti status podatka v podatkovni bazi. Ker je takšna verjetnost precej majhna smo se odločili za ta pristop, kljub ne optimalnosti.
- Pri zaznavanju duplikatov smo se odločili primerjati hash kode html vsebin. Na začetku smo pomotoma hash kodo shranjevali kar v polje html_content, pravo html vsebino pa smo zavrgli. Tako so v bazi strani (tudi s statusom PARSED), ki nimajo določene HTML vsebine. To težavo smo odpravili, je pa to razlog zakaj večina strani v podatkovni bazi nima določenega polja html_content.

6. Zaključek

V programskem jeziku Python smo implementirali spletnega pajka, ki se pomika po domeni gov.si. Sestavljen je iz treh delov: strežnik, odjemalec in baza (Postgre SQL). Taka arhitektura se je izkaza za uspešno izbiro. Pajka smo poganjali 3 dni in zbrali 35188 spletnih strani.