SVEUČILIŠTE U ZAGREBU

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

SEMINAR

Primjena kompleksnih mreža na paketnom stablu Debian linux distribucije

Marijana Novaković

Voditelj: *Mr. Sc. Mile Šikić*

Zagreb, svibanj, 2007.

Sadržaj

Uvod

Ovo je jedan praktičan problem i sam sustav ovisnosti paketa kod Debian distribucije promatramo kao kompleksnu mrežu.

Kompleksne mreže detaljnije su objašnjene u 1. Poglavlju. Za linux distribuciju koristimo Debian (točnije njegovu stable verziju) kao jednu od danas najkorištenijih distribucija sa dobro razvijenim paketnim mehanizmom i kao jednu od distribucija koja ima velik broj varijanti. Ipak svim tim varijantama zajedničko je da koriste pakete s .deb ekstenzijom i sam princip paketnog mehanizma je isti.

U 2. poglavlju prikazujemo način na koji dobivamo trenutan broj paketa, pakete koji o njima ovise i daljnji način obrade, tj. ubacivanje podataka u bazu i njihovo daljnje manipuliranje pomoću kojeg dobivamo graf ovisnosti.

U 3. poglavlju podatke koje smo spremili u bazu podataka dalje obrađujemo i dobivamo graf razdiobe stupnja paketa Debian linux distribucije.

Koristimo alate koji su besplatni i imaju GPL:

- Ruby s dodatcima:
    - Rubygem
    - ActiveRecord gem
    - Hpricot gem
    - MySql gem
- MySql
- Graphviz – program za crtanje grafova

1. Što su kompleksne mreže?

Kompleksne mreže su jedan relativno mlad pojam koji se razvio u 90-im godinama 20. stoljeća kao potreba da se upotpuni pojam klasičnih mreža koje nisu mogle biti u potpunosti upotrijebljene za modeliranje podataka iz stvarnog svijeta. Za razliku od klasičnih mreža, kompleksne mreže imaju kompleksniju distribuciju tj. razdioba veza pri vrhu je složenija od **Poissonove** distribucije i sama struktura je složenija nego kod **Erdos Reny** slučajnih grafova. Sve mreže su u stvarnom svijetu kompleksne i obično imaju distribucije sa debelim repom (*engl.* ***fat tailed degree distribution***).

1.1 Parametri kompleksne mreže

Svaka mreža, pa i kompleksna, definirana je pomoću parametara:

- Distribucija stupnjeva (*engl.* ***degree distribution***)
- Koeficijent grupiranja za cijelu mrežu (*engl.* ***clustering***)
- Najkraći put između pojedinih vrhova
- Promjer mreže
- Funkcija raspodjele broja bridova po pojedinom vrhu
- Zajednica unutar mreže (*engl.* ***community***)
- Najveća komponenta

Za analizu kompleksnih mreža uvijek se na početku koriste prva tri parametra pa su oni detaljnije objašnjeni.

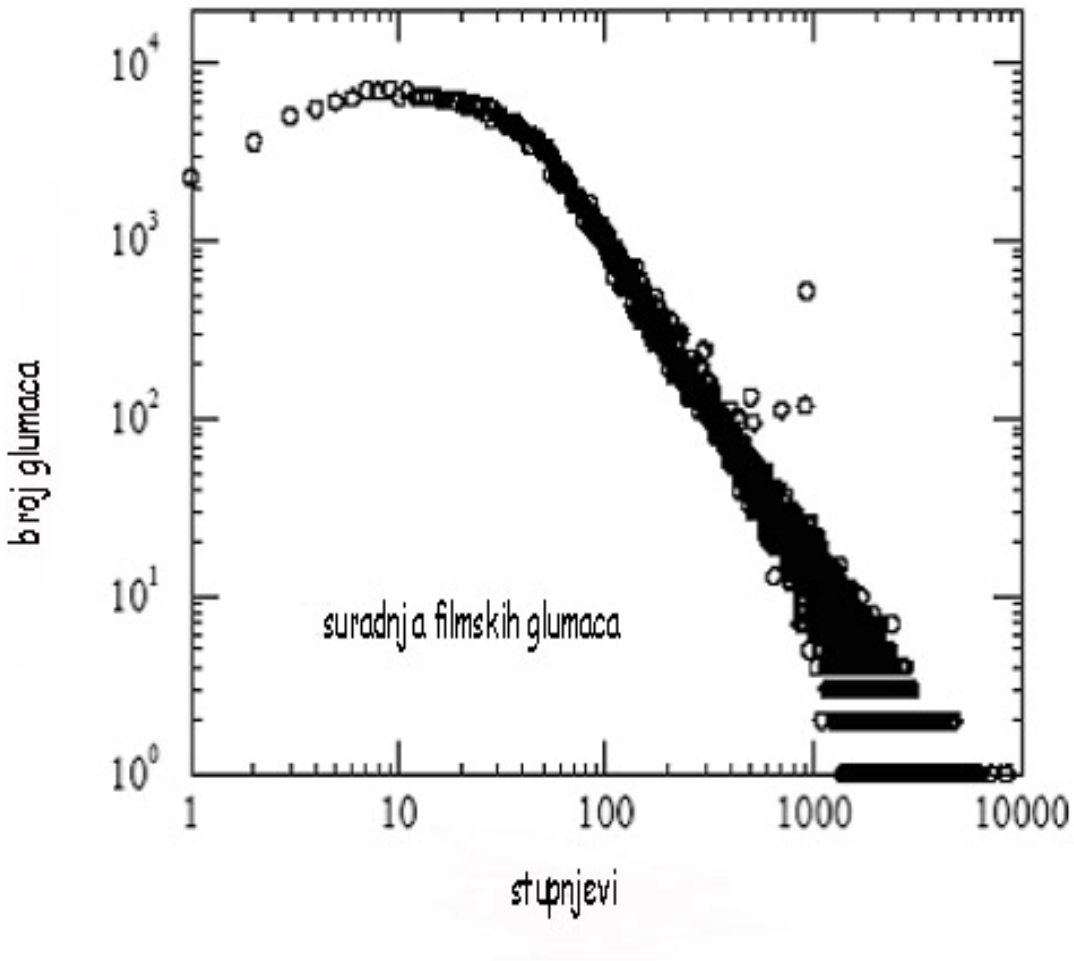
1.1.1 Distribucija stupnjeva

Prvi put pojam distribucija stupnjeva upotrijebili su **Paul Erdős** i **Alfréd Rényi** u svojem radu o slučajnim grafovima. Od tada taj pojam se često koristi za opisivanje topologije kompleksnih mreža. Za svaki vrh najbitniji je njegov stupanj ***k*** tj. broj bridova koji su povezani na taj vrh. U našem slučaju razmatramo usmjereni graf koji osim osnovnog ima i *ulazni (engl.* ***in-degree****)* i *izlazni (****out-degree****)* stupanj. Poznavajući stupanj distribucije svakog vrha, možemo izračunati ukupnu distribuciju stupnjeva mreže. Formula koja opisuje distribuciju stupnjeva glasi:

$$p(k) = \sum_{v \in V \mid \deg(v)=k} 1$$

gdje je ***v*** definiran kao jedan element skupa **V**. Skup **V** je skup svih vrhova, a ***deg(v)*** je definiran kao stupanj vrha.

Na «**Slika 1**. *Razdioba stupnja suradnje filmskih glumaca*» vidimo primjer eksponencijalne razdiobe stupnjeva koja je karakteristična za mreže iz realnog svijeta.Takve forme najčešće imaju «**dugačak rep**».



Slika 1. Razdioba stupnja suradnje filmskih glumaca

1.1.2 Koeficijent grupiranja

Taj pojam su uveli Steven Stroganz i Duncan J. Watts 1998 g. kako bi definirali da li graf spada ili ne u definiciju «**malog svijeta**». Mali svijet je u biti podklasa kompleksnih mreža.

Da bi mogli definirati koeficijent grupiranja, definirajmo pojam susjedstva N za vrh v_i:

$$N_i = \{v_j\} : e_{ij} \in E.$$

Pojam koeficijenta grupiranja definiramo formulom:

$$C_i = \frac{|\{e_{jk}\}|}{k_i(k_i-1)} : v_j, v_k \in N_i, e_{jk} \in E.$$

gdje je Ci koeficijent grupiranja za vrh v_i, a on sam je proporcionalan broju veza između vrhova unutar svoj susjedstva Ni gdje je $k_i(k_i - 1)$ broj veza koji bi mogao postojati unutar susjedstva.

Koeficijent grupiranja za cijeli sustav definira se kao srednja vrijednost koeficijenta grupiranja za svaki vrh:

$$\overline{C} = \frac{1}{n}\sum_{i=1}^{n} C_i.$$

1.1.3 Najkraći put između pojedinih vrhova

Najkraći put (*engl.* ***shortest path***) najčešće koristimo kada želimo optimizirati razne rute. Ovisno o vrsti mreže, najkraći put možemo izračunati na nekoliko načina – pomoću raznih algoritama. Dva algoritma koja se najčešće koriste su:

- **Pretraživanje u širinu** (*engl.* ***breadth first search***)
- **Dijkstrin algoritam**

2. Postupak prikupljanja paketa

Kao referentnu stranicu uzimamo „**http://packages.debian.org**", točnije njihovu potkategoriju sa svim paketima za stable verziju. Program se sastoji od 3 datoteke: **main.rb**, **db.rb** i **graphgen.rb**. Prva datoteka sadrži glavnu funkciju koja skida informacije o paketima, druga datoteka služi za njihovo spremanje u bazu podataka, a treća za generiranje podataka za graf. Prikažimo i objasnimo pseudokod svake svake datoteke pojedinačno. Source kod svih triju datoteka se nalazi u prilogu.

2.1 Main.rb

1. **Korak** – stvaranje liste paketa i spremanje u **tmp.html:**

```
Ukoliko datoteka ne postoji {
        Stvori tmp.html datoteku
        Učitavaj listu paketa pomoću BASE_URL i PKG_URL
        Zatvori datoteku
}
Ako postoji, ispiši da je lista paketa već kreirana.
```

2. **Korak** – otvaranje **tmp.html** i pomoću Hpricota (HTML/XML parser) dijelimo ju na komade:

```
Otvori privremenu datoteku
Pretvori datoteku u Hpricot objekt
Otvori datafile datoteku
Pretraži svaki A element unutar DT elementa (tako
prepoznajemo paket) do kraja dokumenta {
        Upiši ime paketa i njegov URL u datafile
}
        Zatvori datafile datoteku
```

3. **Korak** – Sređivanje paketa za dodavanje u bazu:

```
Otvori datafile datoteku i čitaj ju red po red {
        Podijeli svaku liniju u ime paketa i url do istog
        Pozovi funkciju za definiranje  dubine
        Ukoliko lista ovisnosti nije prazna {
                Za svaki paket {
                        Pronađi ga u bazi ili ga kreiraj u bazi
```

```
                    Dodaj paket u listu ovisnosti
                }
            }
        }
```

2.2 Db.rb

1. **Korak** – Definiramo osnovne informacije za spajanje na bazu**:**
   - Koju bazu koristimo (MySql)
   - Host
   - Ime baze
   - Korisničko ime
   - Lozinku

2. **Korak**

   Definiraj parent – child odnos između paketa

   Međuovisnosti **spremi** u tablicu

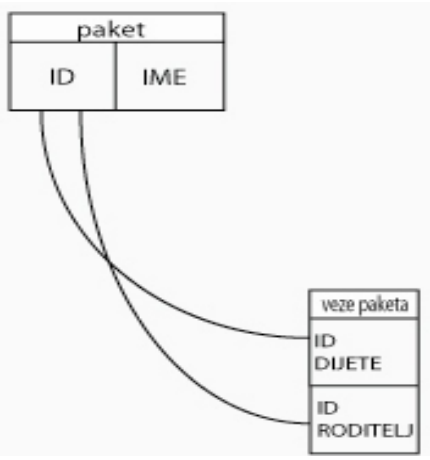
   Definiraj funkcije koje koristimo za dobivanje inputa za crtanje grafa u Graphvizu

Ukratko ponavljamo što smo napravili do sad:

- Skinuli smo listu paketa sa debianove web stranice u privremeni HTML dokument na disku
- Otvorili smo tu istu listu i pretvorili smo ju u Hpricot objekt. Hpricot je HTML parser koji pomoću Xpath expressiona traži određene HTML elemente u našem dokumentu
- U tim HTML elementima se nalazi ime paketa i link do istog i njih spremamo u običnu .dat datoteku
- Čitamo liniju po liniju iz .dat datoteke.
- Svaku pročitanu liniju pretvorili smo opet u Hpricot objekt. Pomoću linka paketa dolazimo do njegovih dependencija koje također zapišemo
- Također provjerava se da se isti paket ne zapiše dvaput u bazi

Jedna od loših strana Hpricota je što zauzima dosta memorije, no naspram ostalih načina najbrži je.

Primjer izgleda baze:



Na „**Slika 2**. Primjer izgleda baze" se vidi da imamo 2 tablice. Svaka tablica sadrži dva elementa.

Prva tablica je tablica „**paket**" koja sadrži ID. On je *primary key*, jednoznačno označuje svaki paket. Ime je ovdje samo opisni atribut.

Druga tablica je „**veze paketa**" koja sadrži ID DIJETE i ID RODITELJ. Oba elementa su *foreign key*. Npr. ako želimo znati kako se paket zove, to moramo provjeriti u „**paket**" tablici.

Slika 2. Primjer izgleda baze

2.3 Graphgen.rb

```
Pronađi paket koji je definiran u argumentu funkcije pri pozivu
Pripremi .dot file za upis podataka
Zapiši u datoteku svaki paket o kojem ovisi zadani paket
Zatvori blok u datoteci
Zatvori datoteku
```

Graphgen pokrećemo naredbom:

ruby graphgen.rb ime_paketa

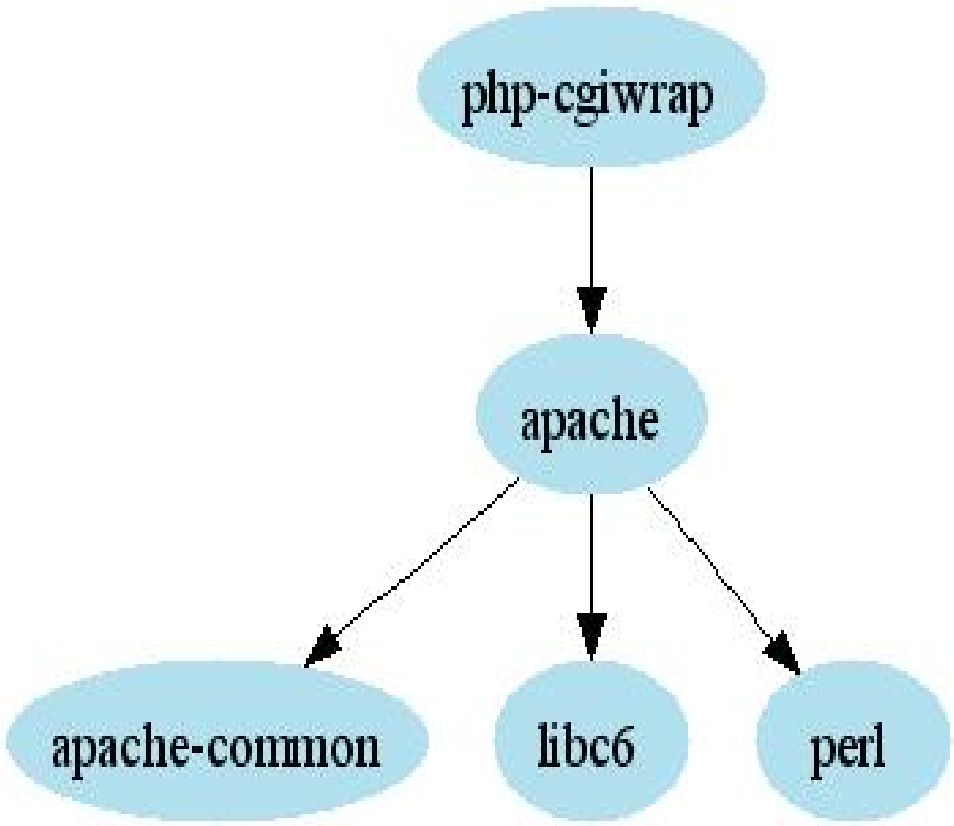
Graphgen zapiše ime_paketa.dot koji služi graphvizu kao input file. Pomoću njega, graphviz crta graf.

Uzmimo primjer **apache** paketa. Graphgen tvori datoteku koja kada se otvori izgleda:

```
digraph unix {
node [color=lightblue2, style=filled];
"apache" -> "apache-common";
```

```
"apache" -> "libc6";
"apache" -> "perl";
"php-cgiwrap" -> "apache";
}
```

Iz te datoteke, graphviz tvori sliku koja izgleda ovako:



Slika 3. Prikaz ovisnih paketa o paketu apache

Iz „**Slika 3.** Prikaz ovisnih paketa o paketu apache" je vidljivo o kojim sve paketima ovisi paket **php-cgiwrap**.

Više o Graphviz programu možete naučiti na „www.graphviz.org".

Nakon što smo prikupili podatke u bazu, slijedi određivanje parametara kojima definiramo našu kompleksnu mrežu. Pomoću već gotovih funkcija izrađenih u PERLu iz baze podataka dobivamo podatke koje spremamo u dvije .gml datoteke. Pomoću tih dviju datoteka dobivamo graf razdiobe stupnjeva paketa Debian linux distribucije.

3. Zaključak

Kako bismo prikazali međuovisnost paketa, upotrijebili smo 3 datoteke:

- **main.rb**
- **db.rb**
- **graphgen.rb**

Pomoću prve dvije datoteke skupljamo podatke, obrađujemo ih kako bi ih lakše organizirali u bazi i onda ih spremamo u bazu. Daljnjim, već definiranim skriptama, dobivamo graf razdiobe stupnjeva paketa Debian linux distribucije. Za one koji žele znati više, u dodatku su sadržane sve tri datoteke sa komentiranim kodom.

Jedna od dobrih strana ovakvog pristupa jest i to što se podacima može lako i jednostavno manipulirati jer su sadržani u bazi podataka.

Kod se može napisati u bilo kojem programskom i skriptnom jeziku. Autorica seminara izabrala je Ruby po osobnoj referenci, no umjesto Ruby-ja moglo se upotrijebiti nešto drugo. Isto vrijedi i za odabir programa za crtanje grafa.

Sama Debian distribucija se najčešće koristi kao serverska distribucija, znači za velik broj servera čije održavanje i *upgrade* se najčešće koristi automatski.

Svi programi koji su upotrijebljeni su dostupni online u svojoj besplatnoj verziji i imaju veliki *support community*.

4. Literatura

[1] S. N. DOROGOVTSEV, J. F. F. MENDES: *The shortest path to complex networks*; arXiv:cond-mat/0404593 v4, 24. srpanj 2004.

[2] W. BACHNIK, S. SZYMCZYK, P. LESZCZYNSKI: *Quantitive and sociological analysis of blog networks*; arXiv:physics/0506051 v1, 07. lipanj 2005.

[3] M. E. J. NEWMAN: *The structure and function of complex networks*; arXiv:cond-mat/0303516 v1, 25. ožujak 2003.

5. Sažetak, ključne riječi

Danas je upotreba klasičnih i kompleksnih veza nešto što se podrazumjeva kao nužnost u istraživanju i analiziranju podataka u bilo kojoj grani znanosti. Za primjer primjene kompleksne mreže u ovom seminaru je uzeto paketno stablo Debian linux distribucije. Kroz seminar je objašnjeno dobivanje i obrada podataka u cilju definiranja kompleksne mreže:

- povlačenje podataka sa službene Debian stranice
- pohrana podataka u bazu
- definiranje parametara mreže iz baze podataka
- formiranje mreže

Tri bitna pojma koja se provlače kroz seminar i koji su detaljnije objašnjeni su:

- **Distribucija stupnjeva**
- **Koeficijent grupiranja**
- **Najkraći put između pojedinih vrhova**

Parametre dobijemo prikupljanjem podataka s Debian packages web stranice i njihovom daljnom obradom te spremanjem u bazu podataka. Nakon dobivanja parametara pomoću već definiranih PERL funkcija definiramo kompleksnu mrežu i dobivamo graf.

Ključne riječi: kompleksna mreža, distribucija stupnjeva, grupiranje, najkraći put između pojedinih vrhova