Boris Dragović, Boško Radivojević

FINES – sistem za balansiranje opterećenosti diskova na Linux operativnom sistemu pod TCP/IP protokolom

U ovom radu je opisan sistem za balansiranje opterećenosti diskova na Linux operativnom sistemu (kernel 1.2.13 i viši) pod TCP/IP mrežnim protokolom. Sastoji se iz dva dela, dela koji je implementiran u Linux kernelu i FNSD-a koji obezbeđuje komunikaciju sa ostalim računarima povezanim u sistem (traženje i prebacivanje fajlova). Osnovna mana projekta je što brzina rada sistema dierektno zavisi od brzine mreže. Celokupan projekat je rađen na C programskom jeziku.

Ključne reči: balansiranje opterećenosti diskova, Linux kernel, TCP/IP.

1. Uvod

FINES je sistem za balansiranje opterećenosti diskova na Linux operativnom sistemu pod TCP/IP mrežnim protokolom. Osnovni cilj sistema je da obezbedi što bolje iskorišćenje slobodnog prostora na diskovima u mreži. Korisnik nije ograničen fizičkim kapacitetom svog diska, nego mu je na raspolaganju slobodan prostor svih diskova u sistemu. Brzina rada sistema direktno zavisi od brzine rada mreže, jer pri otvaranju nekog fajla koji je samo virtuelno na datoj mašini, zaduženi modul FINES sistema mora da "optrči" mrežu u potrazi za datim fajlom i da ga prenese kako bi korisnik mogao nromalno da radi sa njim.

FINES se sastoji iz dva dela. Prvi deo je implementiran u kernelu i njegova glavna uloga je da presreće pristupe fajl sistemu i obaveštava drugi deo FINES sistema, FNSD (File Name Server Daemon) o tome. FNSD obavlja komunikaciju i razmenu fajlova sa drugim FNSD-ovima na mreži. Pored toga, jedan od njih koji je konfigurisan da bude glavni obavlja i samo balansiranje. Ukoliko se desi da mašina koja ima ulogu glavnog FNSD-a postane nepristupačna, njenu ulogu može preuzeti bilo koja druga mašina u sistemu. U ovakvom slučaju balansiranje se nastavlja,

Boris Dragović (1979), Beograd, Vlajkovićeva 4, učenik 2. razreda V beogradske gimnazije; lynx@galeb.etf.bg.ac.yu

Boško Radivojević (1980), Novi Beograd, Pariske komune 33, učenik 1. razreda IX beogradske gimnazije; bole@bolex.bolex.co.yu

ali podaci mašine koja je imala ulogu glavnog FNSD-a, ostaju nepristupačni, kao i podaci sa svake druge "oborene" mašine.

FNSD pri startovanju, ukoliko nije setovan "main flag", pita prvi FNSD u listi koji je FNSD trenutno glavni. Ukoliko ne dobije odgovor pita sledeći. U slučaju da nema nikakvog odgovora preuzima funkciju glavnog FNSD-a. Svi FNSD-ovi, osim onog koji ima funkciju glavnog, su konektovani na njega. U regularnim vremenskim intervalima konektovani FNSD-ovi šalju glavnom (preko konekcije koja je stalno otvorena) kontrolne upite. Ukoliko nema odgovora od glavnog FNSD-a, prvi koji je otkrio ne prisustvo glavnog FNSD-a, preuzima ulogu glavnog i svim FNSD-ovima u sistemu šalje obaveštenje o promeni i traži potrebne podatke (spisak fajlova, dužine i sl.). Tada se konekcije sa prethodnim glavnim FNSD-om raskidaju i upostavljaju se sa novim glavnim FNSD-om. Kada prethodni glavni FNSD otkrije da više nema stalnih konekcija do njega, ponaša se kao da je upravo startovan.

Kontrolne poruke su dovoljno kratke da neće ometati protok podataka kroz mrežu, niti usporavati rad FINES sistema.

2. File Name Server Daemon

FNSD je program koji radi kao pozadinski proces i koji obavlja sve potrebne prenose fajlova, kao i samu optimizaciju opterećenosti. FNSD zauzima određeni port (po "default" konfiguraciji to je port 2020) na kome čeka konekcije. Pri svakoj konekciji na FNSD, proces se klonira, roditelj-proces nastavlja da čeka konekcije, a potomak-proces obavlja komunikaciju sa klijentom koji se konektovao, tako da je teoretski moguć veliki broj istovremenih konekcija.

Svaka konekcija između dva FNSD-a počinje obaveznim predstavljanjem. Predstavljanje podrazumeva da "klijent FNSD" pošalje svoju pristupnu šifru. Tada "server FNSD" proverava da li IP adresa sa koje dolazi "klijent FNSD" odgovara šifri koju je on poslao. Ukoliko se šifra i IP adresa ne slažu, "server FNSD" prekida konekciju uz odgovarajuću poruku, kako bi "klijent FNSD" mogao da "obavesti" administratora o problemu upisujući u odgovarajući log fajl. Svaki FNSD u FINES sistemu je u isto vreme i server i klijent, zavisno od toga da li je primio konekciju ili se konektovao na neki drugi FNSD. Svi FNSD-ovi drže tabele fajlova mašina koji su kod njih, kako bi, ukoliko dođe do zahteva za proveru postojanja datog fajla, pretraživanje bilo brže.

Konektovani FNSD ili "klijent FNSD" može tražiti prenos fajla kome se pristupa, a koji je nekim prethodnim balansiranjem prebačen na neku drugu mašinu. Kada se pošalje komanda, server FNSD pretražuje tabelu fajlova konektovane mašine, koji su fizički kod njega i ukoliko postoji

traženi fajl se prenosi. Prvo se prenosi "stat" struktura iz koje klijent čita dužinu fajla i sve potrebne podatke (mod, vlasnika, grupu, vreme poslednjeg pristupa i slično), a zatim prenosi i sam fajl, koga prosleđuje do procesa koji je pokušao da otvori fajl. Klijent FNSD može poslati fajl, ali je potrebno specifirati "čiji" je taj fajl. To je potrebno jer prilikom balansiranja dolazi do potrebe da fajl mašine A, koji je na mašini B, pređe na mašinu C i da tamo bude registrovan kao fajl mašine A, a ne mašine B sa koje je prebačen. Pored osnovnih "put" i "get" mogućnosti, tu su i "have" i "search". "Have" komanda ispituje da li za konektovanu mašinu postoji dati fajl. "Search" komanda traži po svim FNSD-ovima u sistemu određeni fajl. Tu komandu FNSD-u zadaje kernel, kada dođe do zahteva za određenim fajlom. Kako neke operacije nad fajlovima ne zahtevaju ceo fajl "pred sobom", nego samo "stat" strukturu tog fajla (listanje direktorijuma i sl.), omogućeno je menjanje i slanje "stat" strukture određenog fajla.

FNSD je pravljen tako da bude veoma konfigurabilan, kako bi bio što fleksibilniji pri instalaciji i što efikasniji na određenoj mašini. Svaki fajl i parametar (koji nije privremen) može se definisati. Putanja do glavnog konfiguracion fajl se može definisati pomoću odgovarajućeg "switch"-a.

Identifikatori glavnog konfiguracionog fajla (fnsd.config):

KILLCODE	Šifra za prinudno isključivanje FNSD-a. Ukoliko je identifikator CRYPT true, KILLCODE je kriptovana šifra, standardnim UNIX kript algoritmom
CRYPT	Ukoliko počinje sa Y, CRYPT je true, inače je false
AUTHFILE	Fajl za proveru autentičnosti drugih FNSD-ova Strkutra AUTHFILE-a: <*šifra*>:<*file*>:<*IP*> *šifra* - šifra koja se očekuje pri predstavljanju *file* - Fajl u kome se nalazi da li određeni FNSD može pistati, čitati, putanja gde se čuvaju njegovi fajlovi *IP* - Bazična IP adresa FNSD-a
IMAGEFILE	Fajl u kome se čuva tačna lista fajlova na / fajl sistemu. Sortirani su po dužini i to po opadajućem redosledu, kako bi pretraživanje bilo brže (binarno pretraživanje)
MYPASS	Password lokalnog FNSD-a, služi za proveru autentičnosti sa ostalim FNSD-ovima

LIST	Ime fajla koji sadrži listu drugih FNSD-ova u FINES sistemu. Stuktura LIST fajla: <*IP*>:<*PORT*> *IP* - IP adresa drugog FNSD-a *PORT* - Port na kome je drugi FNSD
PORT	Port koji zauzima FNSD
MAIN	Ukoliko je setovan, mašina ima ulogu glavnog FNSD-a

3. Algoritam za balansiranje

Kao što je već napomenuto, samo balansiranje se vrši isključivo na jednoj mašini. Balansiranje obavlja FNSD u trenutku kada dobije informaciju da je pristupljeno fajl sistemu, odnosno kada se neki fajl zatvori. Tada FNSD ažurira listu fajlova date mašine i predaje algoritmu za balansiranje liste fajlova svih diskova. Algoritam tada pravi složenu dinamičku strukturu podataka koja se sastoji iz dvostruko povezane liste, svaki čvor predstavlja jedan od diskova i sadrži jednostruko povezanu, sortiranu listu veličina fajlova za dati disk. Zatim pokušava da nađe najoptimalniji raspored fajlova (dužine fajlova), koji bi trebalo da budu prebačeni na neki drugi računar kako bi se postigla što bolja izbalansiranost. Algoritmu je moguće predati još jedan parametar – toleranciju, kako bi se ubrzao proces balansiranja. Tolerancija je dozvoljeno odstupanje od apsolutne izbalansiranosti koju definiše sam korisnik. Samim definisanjem tolerancije povećava se broj kombinacija za balansiranje čime se i algoritam znatno ubrzava. Od parametara, algoritmu se prosleđuju i kapaciteti diskova, kako ne bi došlo do slučaja da na jedan disk treba prebaciti više nego što je moguće.

Kao izlazne podatke algoritam FNSD-u predaje vrednosti dužina fajlova koji treba da budu prebačeni i identifikatore računara na kojima se fizički nalaze i na koje treba prebaciti svaki od datih fajlova. Algoritam za balansiranje je predstavnjen pseudo kodom. Gore pomenuta dinamička struktura podataka je već kreirana i predata algoritmu (početni čvor je "firstdisk" i on predstavlja prvi disk u sistemu).

```
nadji_srednju_vrednost_za_sve_diskove
trenutnidisk = firstdisk
dok je trenutnidisk razlicit od NULL
{
  diskpointer = trenutnidisk->sledeci
  dok je diskpointer razlicit od NULL
  {
```

```
   treba_prebaciti = srednja_vrednost - diskpointer
   -> opterecenje
    vrednostpointer = diskpointer -> lista_vradnosti
dok je vrednostpointer razlicit od NULL i
    treba_prebaciti razlicito od nule
    {
     ako je vrednostpointer-velicina manje ili
     jednako od treba_prebaciti
      {
       treba_prebaciti smanji za vrednostpointer
       ->vrednost
       prebaci vrednostpointer-vrednost sa
       diskpointer na trenutnidisk
      }
     vrednostpointer = vrednostpointer -> sledeci
    }
    diskpointer = diskponter-sledeci
  }
  trenutnidisk = trenutnidisk-sledeci;
}
```

4. Promene u kernelu

Promene u kernelu su bile neophodne zbog potrebe presretanja svih pristupa fajl sistemu. Presretanje pristupa fajl sistemu neophodno je zbog toga što će korisnik možda želeti da pristupi fajlu koji je balansiranjem prebačen na neki drugi računar ili pak zbog novog opterećivanja diska snimanjem fajla ili dodavanjem podataka u neki od fajlova. U ovakvim slučajevima dužnost rutine u kernelu je da obavesti FNSD koji će dalje u zavisnosti od onoga šta se dešava u fajl sistemu obavljati svoj deo posla (balansiranje ili pretragu po drugim FNSD-ovima i prenošenje fajla). Rutina u kernelu ostvaruje međuprocesnu komunikaciju sa FNSD-om preko kreiranja *socket*-a i konektovanja na fiksni port FNSD-a. Izmene su napravljene u */usr/src/linux/fs* direktorijumu gde se nalaze izvorni kodovi rutina koje su zadužene za rad sa fajl sistemom.

Kod, koji je dodat u kernel, je potpuno kompatibilian sa svim fajl sistemima koje *Linux* podržava, znači neće praviti probleme pri prelasku sa jednog na drugi (npr. sa *minix* na *ext2*).

5. Pomoćni programi

Kada je potrebno jednu mašinu izdvojiti iz FINES sistema, a da to izdvajanje bude bez gubitka podataka, koristi se pomoćni program nazvan *collector*.

Collector prvo skuplja i sabira sve dužine fajlova koji pripadaju mašini. Ukoliko fajlovi date mašine zauzimaju više nego što fizički može

da stane na mašinu (naravno, vodi se računa o fajl sistemima) prekida se rad programa i ispisuje se poruka o grešci. Ukoliko je sve u redu, odnosno sve što pripada jednom fajl sistemu može fizički i da stane, *collector* obaveštava glavni FNSD o svom izlasku iz sistema. Glavni FNSD tada obaveštava sve ostale FNSD-ove u sistemu da dotični računar više nije u sistemu. Tada je FNSD-ovi u FINES sitemu izbacuju iz liste mašina povezanih u sistem. Posle toga, *collector* vraća računarima fajlove koji su na računaru koji se izdvaja. Onda skuplja sve svoje fajlove sa ostalih FNSD-ova. Nakon završene operacije, prekida se rad FNSD-a i obaveštava administrator da je potrebno izbaciti startovanje FNSD-a u start-up sekvenci.

Pored *collectora*, postoji i program za prikazivanje trenutnog stanja u sistemu, npr. zauzetost diskova, ukupni prostor, odstupanje od apsolutne izbalansiranosti itd.

Literatura

[1] Beck, Bohme, Dziadzk, Kunitz, Magnus, Verworner. 1996. *Linux kernel internals*.

[2] Stevens, Richard W. 1990. *UNIX Network Programming*.

Boris Dragović, Boško Radivojević

FINES – System for Load Balancing Over TCP/IP Family of Network Protocols for Systems Based on Linux Operating System

FINES is program for load balancing over TCP/IP family of network protocols for systems based on Linux operating system kernel 1.2.13 or higher. It consists of two parts, first one is implemented in Linux kernel and the second one is File Name Server Deamon, which ensures communication with File Name Server Deamons on other machines, does load balancing computations and transfers files over network. Whole project has been done in C programming language.

Keywords: load balancing, Linux kernel, TCP/IP