Korištenje DRBD i Heartbeat tehnologije za postizanje stalne dostupnosti servera baziranih na Linux operativnom sistemu

Budimir Kovačević
Univerzitet u Istočnom Sarajevu
Elektrotehnički fakultet
Istočno Sarajevo, Republika Srpska, BIH
budimir.kovacevic@etf.unssa.rs.ba

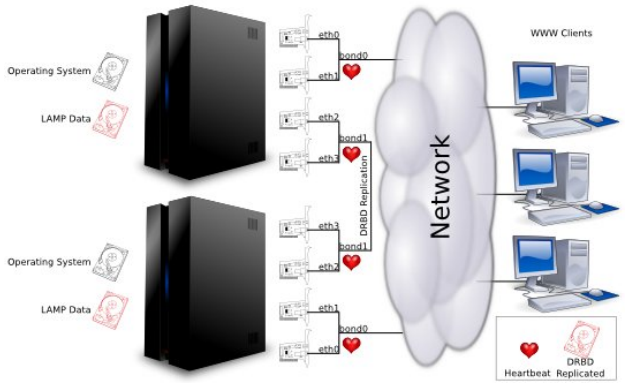
Sadržaj—**Jedan od značajnih problema internet baziranih ili bilo kojih poslovnih aplikacija koje se izvršavaju na nekom serveru je kako obezbjediti njihov neprekidan rad i stalnu dostupnost. Da bi se ovo postiglo potrebno je obezbjediti redudantnost svih komponenti koje mogu izazvati prekid u radu cijelog sistema. Cilj ovog rada je da pokaže kako je na veoma praktičan i ekonomičan način moguće obezbjediti stalnu dostupnost servera. Predstavljeno je rješenje koje koristi DRBD i heartbeat tehnologiju.**

Ključne riječi-DRBD; heartbeat; redudantnost; high availability; HA klasterovanje; RAID; mirroring; preslikavanje;

I. UVOD

Za internet stranice, internet bazirane aplikacije, poslovne aplikacije i druge servise veoma je važno da njihov rad bude neprekidan i da se veoma brzo oporavljaju od grešaka. Svaki prekid ili kašnjenje u radu servisa izaziva nezadovoljstvo kod korisnika te usluge. Korisnici danas očekuju 24 x 7 dostupnost čak i za servise koji nisu od presudnog značaja. HA (eng. *High availability*) klasterovanje omogućava serverima da se brzo oporave od greške i nastave sa radom. [1]

Da bi se obezbjedila potpuna redudantnost potrebno je da se osigura da u cijelom sistemu ne postoji jedna tačka čija neispravnost može narušiti rad cijelog sistema.



Slika 1. Šema HA klasterovanja

U ovom radu ideja je da se prikaže rješenje koje predstavlja najbolji odnos cijene i kvaliteta. Takođe je navedeno i rješenje koje omogućava da sistem nastavi sa radom, čak i nakon katastrofa kao što su poplave, požari, zemljotresi korištenjem asinhronog preslikavanja. Na Sl. 1, prikazana je šema HA klasterovanja. [2] Detalji svih veza biće obrađeni dalje u radu.

II. IDENTIFIKOVANJE I ELIMINACIJA TAČAKA U KOJIMA MOŽE DOĆI DO KVARA

Da bi se obezbijedio neprekidan rad sistema, potrebno je da postoji redudantnost na svim nivoima. Većina hardverskih kvarova koji nastaju prevazilaze se korištenjem potpuno istih komponenti koje na sebe preuzimaju funkciju uređaja koji je neispravan. Posebnu pažnju treba posvetiti sljedećim komponentama:

1) *Diskovi*
2) *Napajanja*
3) *Hlađenja*
4) *Mrežne kartice*

Diskovi predstavljaju najslabije tačke servera. Mali mehanički dijelovi unutar diskova su vrlo osjetljivi na toplotu i vibracije. Takođe, jedan od problema sa diskovima je da je njihov vijek trajanja ograničen. Diskove je potrebno postaviti u RAID 1 (preslikavanje) ili neki drugi RAID u slučaju da postoji više diskova. U slučaju da se koristi RAID 1, preporučuje se da drugi disk bude od drugog proizvođača.

Napajanje servera predstavlja sljedeći prioritet. Ovaj problem se rješava postavljanjem redudantnog napajanja sistema koje je vezano na drugu napojnu liniju. Nakon gubitka jedne linije napajanja server neprekidno nastavlja da radi. Pored toga, preporučuje se da se koristi rezervno napajanje preko baterije (eng. *Uninterruptible Power Supply - UPS*), kako bi sistem nastavio nesmetano da radi i nakon gubitka napajanja na obje linije.

Aktivno hlađenje je neophodno za procesore i druge komponente servera koje se griju prilikom rada. Postavljanje dodatnih ventilatora u mnogome smanjuje opasnost od pregrijavanja komponenti. Današnji serveri omogućavaju da se kontroliše brzina okretanja ventilatora i da upozoravaju

administratora sistema kada brzina okretanja ventilatora padne ispod određenog praga.

Rijedak, ali mogući problem je greška prilikom spajanja na *mrežu* koja nastaje usljed prestanka rada porta na sviču ili cijelog sviča. Povezivanjem više mrežnih interfejsa servera na svič rješava se problem koji nastaje zbog neispravnoga porta. Ovim se ne prevazilazi problem nastao usljed neispravnosti cijelog sviča. Tako da je veoma dobro rješenje da se server poveže na dva sviča. Ovakvo povezivanje zahtjeva ISL (eng. *Inter Switch Link*) vezu između svičeva i da oni podržavaju STP (eng. *Spaning Tree Protocol*) protokol koji spriječava stvaranje petlji. Upravljivi Cisco Catalyst svič sa rezervnim napajanjem je sasvim dobro rješenje.

III. HA KLASTEROVANJE

Korištenjem HA klasterovanja povećava se vrijeme raspoloživosti uređaja a smanjuje se vrijeme koje je potrebno da se sistem oporavi od nastale greške. Stalna dostupnost servisa smanjuje nezadovoljstvo korisnika istog. Danas postoje specijalizovana rješenja koja su prilagođena odgovarajućem hardveru. Ova rješenja zahtijevaju korištenje skupih hardverskih i softverskih komponenti industrijskih lidera kao što su IBM, HP, SUN, VMware, i drugi.

Postoje tri metode HA klasterovanja:

1) *HAC (*eng. *High Performance Clustering)*
2) *HPC (*eng. *High Performance Computing Clustering)*
3) *LL (*eng. *Load Leveling Clustering – Load Balancing)*

U radu je fokus stavljen na HAC metodu. Veoma je važno razumjeti kako se aplikacije i podaci distribuiraju kroz više servera u klasteru.

Postoji active/active klasterovanje (oba čvora su aktivna i dijele ukupno opterećenje) i active/passive klasterovanje (samo je jedan čvor produktivan). Kod prvog tipa klasterovanja treba voditi računa da veliki broj aplikacija ne podržava paralelelizam u radu. Konfiguracija ovog tipa je mnogo složenija i samim tim je skuplji za održavanje. Prednosti acitve/passive tipa su što je dosta lakši za konfiguraciju a samim tim mu je i održavanje jeftinije i ne mora se voditi računa o tome da li aplikacije podržavaju paralelizam u radu. Nedostatak ovog sistema je što su računarski resursi drugog servera neiskorišteni.

Osnovni dio svakog HA klastera je softver koji nadgleda dostupnost servera u klasteru. Postoje brojna softverska rješenja, međutim većina njih su veoma skupi. Komercijalni primjeri ovih softvera su HACMP od IBM-a (dostupan za IBM, AIX i Linux), HP Service Guard (dostupan za HPUX i Linux).

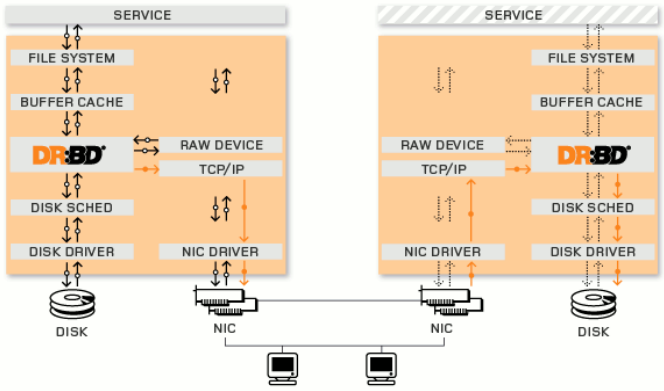
Osnovna funkcionalnost ovog softvera je nadgledanje partnerskog čvora i preuzimanje njegovih funkcija u slučaju da se detektuje kvar. Svaki čvor korištenjem heartbeata ili pinga provjerava da li je partnerski čvor operativan. Ako udaljeni čvor nije operativan, neće odgovoriti potvrdno na ove poruke, a partnerski čvor će u ovom slučaju pokrenuti proceduru za preuzimanje njegovih servisa. Da bi se prebacivanje izvelo potrebno je da čvor preuzme IP adresu, podigne fajl sistem i pokrene potrebne aplikacije. [3]

IV. DRBD I HEARTBEAT

Ovo rješenje čine DRBD koje čini osnovu HA klastera i Heartbeat kao softver za nadgledanje. Ovo je najpouzdanije besplatno rješenje koje postoji.

A. *DRBD*

DRBD je modul namjenjen za Linux operativne sisteme. Omogućava replikaciju podataka sa lokalnog bloka uređaja na drugi blok uređaja udaljenog čvora. Preslikavanje se ostvaruje preko mreže. Smješta se između drajvera diska i fajl sistema. Za DRBD se može reći da predstavlja RAID 1 servera gdje se preslikavanje vrši preko mreže. [4]



Slika 2. DRBD modul u sistemu

Na Sl. 2 su prikazana dva servera koja čine HA klaster. Prikazane su uobičajene komponente Linux operativnog sistma, TCP/IP i drajveri mrežne kartice. Crnim strelicama prikazan je uobičajeni tok podataka između ovih komponenti. Narandžastim strelicama prikazan je tok podataka, koji se kroz DRBD preslikavaju sa aktivnog na pasivni čvor u HA klasteru. [5]

Prvi nivo ovog softvera predstavlja drajver diska (SCSI ili SATA). Obično su SCSI diskovi na Linux operativnom sistemu označeni sa /dev/sd*, dok su SATA ili PATA diskovi označeni /dev/hd*. Ovi uređaji predstavljaju osnovu DRBD-a. DRBD uređaji označeni su sa /dev/drbd*. Osnovni DRBD uređaji mogu da budu bilo koja vrsta blok uređaja kao što su cijeli diskovi, particije, logički diskovi, i drugi. DRBD uređaji se ponašaju kao bilo koji drugi uređaj u sistemu, što ih čini nezavisnim od aplikacije.

Podaci se preslikavaju u realnom vremenu. Svaki upis na disk, pokreće mrežnu operaciju koja upisuje promjene na disk drugog čvora. Kao rezultat ovoga imamo da svaki čvor ima iste podatke smještene na svojim lokalnim diskovima. DRBD koristi tri različita protokola koji imaju različit nivo pouzdanosti.

Protokol „A“ osigurava potvrdu upisa na lokalni disk primarnoga čvora što je sasvim dovoljno za asinhrono preslikavanje podataka.

Protokol „B“ dodaje novi nivo pouzdanosti koji potvrđuje da je podatak upisan na lokalni disk primarnoga čvora samo ako je udaljeni čvor primio podatke u svoj bafer za upis.

Protokol „C“ je najpouzdaniji protokol. Da su podaci upisani na lokalni disk potvrđuje se tek nakon što oni budu upisani na disk udaljenog čvora. HA klasteri obično koriste protokol „C“ zato što on garantuje da je sistemska transakcija sigurna operacija. Ovaj protokol zahtjeva da propusni opseg mreže bude veliki između ova dva čvora. Protokol „A“ i „B“ se koriste samo ako je propusni opseg između dva čvora ograničen.

Da bi se postigle veoma visoke performanse i velika pouzdanost, čvorovi moraju biti locirani u istoj prostoriji i koristiti protokol „C“. Najbolje je da se koriste odvojeni gigabitni interfejsi za DRBD preslikavanje. [6]

B. *Konfiguracija DRBD-a*

Postoji samo nekoliko stvari koje potrebno podesiti kod DRBD-a. Podešavanje se vrši u konfiguracionom fajlu koji je smješten u /etc/drbd.conf. Potrebno je konfigurisati sledeće:

1) *Osnovni uređaj*
2) *Protokol koji se koristi*
3) *Parametre povezivanja kao što su IP adrese i portovi*

Pretpostavimo da je na oba servera particija koja je dodjeljena DRBD-u označena sa /dev/hda2, onda DRBD možemo da konfigurišemo kao što je prikazano na Sl. 3:

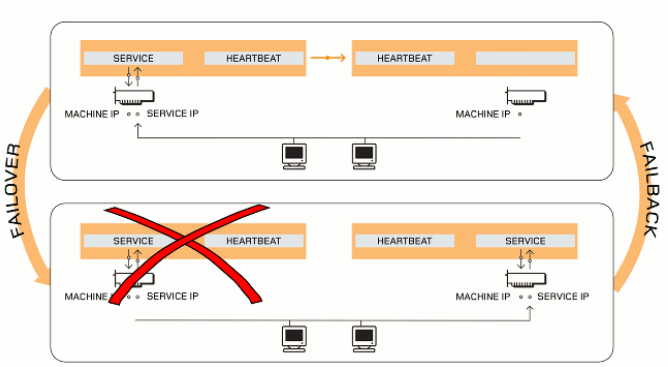
```
resource r0 {
    net {
        protocol C;
    }
        startup {
        degr-wfc-timeout 120; # 2 minuta.
    }
    disk {
        on-io-error detach;
    }
    syncer {
        rate 100M;
        al-extents 2;
    }
    on server1 {
        device /dev/drbd0;
        disk /dev/hda2;
        address 10.0.0.1:7788;
        meta-disk internal;
    }
    on server2 {
        device /dev/drbd0;
        disk /dev/hda2;
        address 10.0.0.2:7788;
        meta-disk internal;
    }
```

Slika 3. Prikaz drbd.conf fajla

Naziv resursa (u ovoj konfiguraciji r0) definiše administrator sistema. Nakon imena, definiše se protokol koji se koristi. U djelu „startup“ definiše se početno vrijeme. Ovo vrijeme predstavlja vrijeme koje je potrebno da server sačeka kako bi se ostvarila konekcija sa partnerskim čvorom. „syncer“ predstavlja maksimalnu brzinu dozvoljenu za sinhronizaciju podataka između dva čvora. Ova brzina zavisi od brzine mreže između dva čvora kao i od brzine diskova. Najvažniji dio konfiguracije predstavlja „on server1“ i „on server2“. Ova imena predstavljaju imena servera i moraju da se poklapaju sa izlazom komande „uname -n“. Potencijalni problem može da bude firewall, tako da treba obezbjediti čvorovima da mogu da komuniciraju preko porta 7788.

C. *Heartbeat*

Heartbeat igra glavnu ulogu kod HA klasterovanja. Ovo je softver koji prati aktivni čvor i čeka da se desi neki kvar. Ako on ne odgovori u vremenu koje je definisano, heartbeat preuzima IP adrese, diskove i servise.



Slika 4. Uloga heartbeat-a

Na Sl. 4 je prikazan klaster u kome je lijevi čvor aktivan i klijenti pristupaju IP adresi i servisima koja se nalazi na lijevom čvoru. IP adresa i servisi mogu da se prebace na desni čvor u bilo kom trenutku od strane administratora ili prebacivanje može da se desi automatski ako dođe do kvara na aktivnom čvoru. Na donjoj slici prikazan je pad aktivnog čvora. Akcija koja se dešava nakon ovoga naziva se *failover*. Suprotan proces, kada se aktivni čvor vrati u ispravno stanje naziva se *failback*. Ako prebacivanje vrši administrator to se naziva *switchover*. [7]

D. *Konfiguracija heartbeat-a*

Postoje tri osnovna fajla koja je potrebno izmjeniti kako bi se izvršilo podešavanje heartbeata: *ha.cf, haresources* i *authkeys*. Svi ovi fajlovi nalaze se u /etc/ha.d direktorijumu. ha.cf sadrži informacije o imenu čvora i heartbeat konekciji. U fajlu haresources čuva se definicija servisa u klasteru. Fajl authkeys koristi se za čuvanje dijeljenog tajnog ključa čime se sprječava neovlaštena komunikacija.

```
auto_failback off
ucast eth0 10.0.0.1
ucast eth1 192.168.0.1
serial /dev/ttyS0

node server1
node server2
```

Slika 5. Prikaz ha.cf fajla

Na Sl. 5 prikazana je skraćena verzija fajla ha.cf, gdje su navedeni parametri koje je potrebno promijeniti u odnosu na podrazumjevane.

Parametar „auto_failback off“ upućuje na to da se neće desiti automatsko prebacivanje na primarni čvor nakon njegovog oporavka. Ovdje imamo server sa dva mrežna interfejsa. Jedan od ova dva interfejsa je povezan direktno na javnu mrežu dok je drugi povezan direktno na drugi server i koristi se samo za replikaciju podataka. Ovdje se upisuju IP adrese interfejsa. *Ucast* označava da se radi o unicast IP adresi. Na kraju konfiguracionog fajla navode se imena čvorova, koja moraju da se podudaraju sa izlazom komande uname –n. Prvi čvor definiše primarni server.

Haresources fajl koristi se da bi definisao fajl sisteme, IP adrese i aplikacije kojima će se upravljati preko klastera. Ovaj fajl mora da bude identičan kod oba čvora. Kod ovog fajla potrebno je promijeniti sljedeće parametre kao na Sl. 6.

```
server1 10.0.0.3 192.168.0.3 \
      drbddisk::r0 \
      Filesystem::/dev/drbd0::/data1::ext3 \
      mysql.server
      apachectl
```

Slika 6. Prikaz haresources fajla

Podatak „server1“ predstavlja primarni čvor i važi za oba čvora. U nastavku se definišu IP adrese. Ove IP adrese su aktivne na primarnom, ali se dijele između oba čvora. Može se definisati neograničen broj IP adresa. Ovdje su definisane dvije IP adrese, jedna za javnu i druga za lokalnu mrežu. Ako se koristi više parametara onda se oni razdvajaju sa dvotačkom. Svi parametri koje treba podesiti se upisuju na jednu liniju ili se razdvajaju korištenjem znaka „\“. Dalje se navodi DRBD uređaj i resurs koji ga koristi. Ispod se nalazi instrukcija, koji bi uređaj trebalo da se podigne, u kojoj tački i koji fajl sistem da koristi. U nastavku je moguće navesti aplikacije koje je potrebno pokrenuti kada se desi preuzimanje nakon greške.

Fajl authkeys sadrži dijeljeni tajni ključ koji je šifrovan odabranim algoritmom. Na Sl. 7 prikazan je sadržaj ovog fajla:

```
auth 1
1 sha1 OvdjeUpisatiLozinkuZaHeartbeat
```

Slika 7. Prikaz drbd.conf fajla u konfiguraciji sa tri čvora

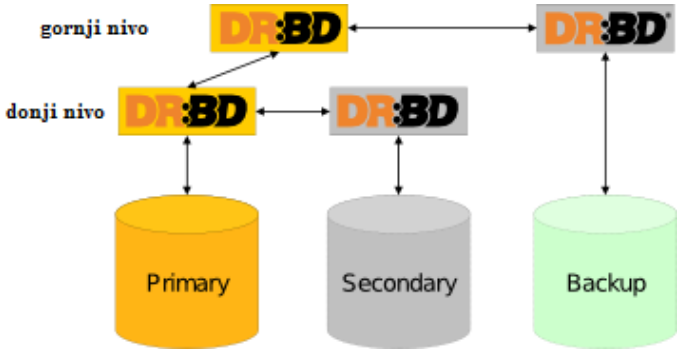
U prvom redu naznačeno je da se koristi ključ sa rednim brojem 1. U drugom redu nalazi se redni broj ključa (1), algoritam kojim se ključ šifruje i ključ. Podržani algoritmi za šifrovanje su *sha1*, *md5* i *crc*. [8]

V. DRBD U SLUČAJU KATASTROFA

Da bi se postigle veoma visoke performanse i velika pouzdanost, čvorovi moraju biti locirani u istoj prostoriji kao što je navedeno gore. U ovom slučaju oba čvora će biti uništena u slučaju katastrofa kao što su požar, poplava i slično i gubitak podataka u ovom slučaju ne može biti spriječen.

A. *Three-way replication*

Da bi se podaci sačuvali potrebno je uraditi „*Three-way replication*“, koja je prikazana na Sl. 8. Ovdje se dodaje treći čvor i na njega se vrši replikacija podataka sa prethodna dva.



Slika 8. Preslikavanje servera kod slučaja da imamo tri čvora

Treći čvor se postavlja na udaljenu lokaciju i koristi protokol „A“ jer se radi asinhrona replikacija dok se između druga dva čvora koristi protokol „C“. Nedostatak protokola „A“ je naveden predhodno u radu i može da dovede do djelimičnog gubitka podataka ali ne velikog. Sinhronizacija podataka sa trećim čvorom može da se radi i na zahtjev, tako što se napravi „*cron*“ koji će ovo da uradi u naznačeno vrijeme, na primjer kada mreža nije opterećena.

B. *Konfiguracija backup servera*

Da bi se ostvario ovaj način replikacije u konfiguracioni fajl drbd.conf čiji je sadržaj prikazana na Sl. 3, potrebno je dodati još jedan resurs kao što je prikazano na Sl. 9:

```
resource r0-U {
      net {
            protocol A;
      }
      stacked-on-top-of r0 {
            device     /dev/drbd10;
            address    78.28.157.10:7788;
      }
      disk {
            on-io-error detach;
      }
      syncer {
            rate 100M;
            al-extents 2;
      }
      on server3 {
            device /dev/drbd10;
            disk /dev/hda6;
            address 78.28.157.11:7788;
            meta-disk internal;
      }
}
```

Slika 9. Prikaz drbd.conf fajla u konfiguraciji sa tri čvora

VI. ZAKLJUČAK

Korištenjem softvera baziranih na DBRD i heartbeat tehnologiji HA klasterovanje je moguće ostvariti u potpunosti, i to besplatno. Ono počinje na nivou operativnog sistema a završava na aplikativnom nivou. Korisnici danas očekuju 24 x 7 dostupnost servisa koje koriste, te je zato neophodno obezbjediti njihov neprekidan rad tako što će se isključiti sve tačke koje mogu da dovedu do prekida u radu.

Nedostatak ovog rješenja je što je ograničeno na Linux operativne sisteme. Ovaj nedostatak i nije veliki ako uzmemo u obzir da većina današnjih servera koristi ovaj operativni sistem za svoje servere.

LITERATURA

[1] Umar Farooq Minhas, Shriram Rajagopalan, Brendan Cully, Ashraf Aboulnaga, Kenneth Salem, Andrew Warfield, "RemusDB: transparent high availability for database system," The VLDB Journal, Online First, 17 Oktober 2012.

[2] HighlyAvailableLAMP, https://help.ubuntu.com/community/HighlyAvailableLAMP

[3] Luc de Louw, „High availability infrastructures for TYPO3 Websites", Proceedings Conference October 05-08, 2006, Karlsruhe, Germany.

[4] Lars Ellenberg, „DRBD 9 & Device-Mapper", Proceedings of Linux-Kongress 2008 October 7-10, 2008, Hamburg, Germany.

[5] www.drbd.org.

[6] Philipp Reisner, „DRBD - Distributed Replicated Block Device", 9th International Linux System Technology Conference, August 12, 2002

[7] http://www.drbd.org/home/what-is-ha/

[8] http://www.linux-ha.org/wiki/Authkeys

ABSTRACT

One of the important problems of internet-based applications or any other business applications executed on a server is how to provide their continuous work and permanent availability. To achieve this, it is necessary to provide the redundancy of all components that might cause an interruption in the work of the entire system. The aim of this paper is to show how the permanent availability of server can be achieved in very practical and economic way. The solution that uses DRBD and heartbeat technology is presented.

THE USAGE OF DRBD AND HEARTBEAT TECHNOLOGY FOR ACHIEVING PERMANENT AVAILABILITY OF SERVERS BASED ON LINUX OPERATING SYSTEM

Budimir Kovacevic