

54. konferencija za
elektroniku, telekomunikacije,
računarstvo, automatiku i
nuklearnu tehniku

2010

ETRAN

Zbornik apstrakta
i program



Organizator
Društvo za ETRAN

Donji Milanovac
7-10. juna 2010.

KORISNIČKI TRANSPARENTNO KREIRANJE REZERVNIH KOPIJA PODATAKA U VIRTUALNOM VMWARE OKRUŽENJU

Vladimir Ćirić, Oliver Vojinović
Elektronski fakultet, Univerzitet u Nišu,
{ciric, oliver}@elfak.ni.ac.rs

Sadržaj – *Razvoj platformi za virtualizaciju kompletnih uređaja uvodi novi stepen slobode pri definisanju načina na koji se kreira rezervna kopija podataka. Cilj ovog rada je projektovanje sistema i implementacija Linux skript programa za kreiranje rezervnih kopija uređaja, na jednom nivou, i korisničkih podataka na drugom nivou, na takav način da se ne prekida i ne ometa rad korisnika sistema. Princip transparentnog kreiranja rezervnih kopija virtualnih servisa biće ilustrovan UML dijagramom. Metod će biti implementiran na VMware platformi VMware skript jezikom. U radu će biti dati rezultati opterećenja resursa u procesu kreiranja rezervnih kopija.*

1. UVOD

Skoro 70% korisnika računara je bar u nekoj meri iskusilo gubitak važnih podataka [1]. Gubitak podataka moguće je sprečiti po ceni redovnog kreiranja rezervnih kopija podataka. U informacionim tehnologijama pojam kreiranja rezervne kopije (eng. *backup*) odnosi se na kreiranje kopije podataka tako da je u slučaju gubitka originalnih podataka podatke moguće vratiti iz sačuvane kopije (eng. *restore* proces) [2]. Primarna namena rezervnih kopija je dvojaka. Prvo, rezervne kopije se koriste za restauraciju celokupnog stanja koje je prethodilo gubitku. Ovaj proces je poznat pod nazivom "restauracija nakon katastrofe" (eng. *disaster recovery*), gde se pojam "katastrofa" vezuje za prirodne nepogode. Druga namena je restauracija manjeg broja slučajno izgubljenih ili oštećenih fajlova, uglavnom zbog greške korisnika [2].

Postoji mnogo metoda koje rešavaju problem kreiranja rezervnih kopija. Sve metode moguće je klasifikovati u dve kategorije: kreiranje kopije na nivou uređaja (eng. *device-based scheme*) i kreiranje kopije na nivou fajlova (eng. *file-based scheme*) [3]. Dok mnoge metode za kreiranje rezervnih kopija zahtevaju stavljanje sistema van upotrebe za vreme kreiranja rezervne kopije, grupa metoda pod nazivom *on-line* metode dozvoljavaju nesmetani rad korisnika za vreme kreiranja kopije. Kod *on-line* procesa kreiranja rezervnih kopija, proces se odvija transparentno za korisnika [3]. Korisnički transparentno kreiranje rezervnih kopija je godinama unazad izazov za istraživače i kompanije zbog problema konzistentnosti podataka koji se može javiti i narušiti čitav koncept [4].

Virtualizacija je proces apstrakcije hardverskog ili softverskog resursa, i primenjuje se u računarstvu još od 1960. godine, kada je kod tadašnjeg Atlas računara i IBM M44/44X primenjen koncept virtualne memorije [5].

Uz razvoj hardverskih platformi koje omogućavaju virtualizaciju kompletnih uređaja sa operativnim sistemom, uređaj se apstrahuje skupom fajlova, čime se daje veći stepen slobode pri definisanju načina kreiranja rezervnih kopija.

Cilj ovog rada je projektovanje sistema i implementacija Linux skripti za kreiranje *online* rezervnih kopija na nivou uređaja u virtualnom okruženju. U radu je predložen metod koji u virtualnom okruženju rešava problem eventualne nekonzistentnosti rezervne kopije prilikom *on-line backup* procesa. Zbog zahtevnosti *online* metoda po pogledu resursa, na jednom nivou će biti razmatran *device-based*, a na drugom nivou *file-based* koncept u VMware virtualnom okruženju [6]. Predloženi metod kreiranja rezervnih kopija virtualnih uređaja biće ilustrovan UML dijagramom. Metod će biti implementiran na VMware platformi VMware skript jezikom [7]. U radu će biti dati rezultati opterećenja resursa u procesu kreiranja rezervnih kopija.

2. KLASIFIKACIJA METODA ZA KREIRANJE REZERVNIH KOPIJA

U ovom poglavlju daćemo kratak pregled postojećih tehnika za kreiranje rezervnih kopija.

Najjednostavniji način zaštite sistema od otkaza diskova ili oštećenja fajlova je kreiranje kopije celokupnog sadržaja fajl sistema na *backup* uređaj. Rezultujuća arhiva naziva se potpuni *backup*. U slučaju gubitka fajl sistema usled oštećenja diska, fajl sistem je moguće restaurirati iz arhive na novi disk. Individualni fajlovi takođe mogu biti restaurirani. Potpuni *backup* ima dva važna nedostatka: arhiviranje i rekonstrukcija celog fajl sistema je spora, a čuvanje rezervne kopije celog sistema zahteva veliki kapacitet *backup* medijuma [3].

Brži *backup* postiže se tzv inkrementalnim procesom kreiranja rezervnih kopija, koji kreira kopije samo onih fajlova koji su modifikovani od poslednjeg kreiranja rezervne kopije podataka [3, 4]. Inkrementalni *backup* smanjuje obim zadatka kreiranja rezervne kopije, jer se samo manji deo fajlova promeni u toku dana.

Kreiranje kopije na nivou fizičkog uređaja (*device-based*) ignoriše strukturu fajl sistema prilikom kreiranja kopije. Ovo značajno može ubrzati kopiranje zbog smanjenja skakanja sa bloka na blok, a samim tim i smanjenja broja mehaničkih pomeraja glava diska. Međutim, ovaj pristup značajno komplikuje restauraciju pojedinih

fajlova [3].

Korisnički transparentno kreiranje rezervnih kopija (*online backup*) je kompleksan zadatak zbog problema konzistentnosti podataka. U [4] razmatrani su problemi koji mogu nastati kod *on-line backup* procesa korišćenjem UNIX *tar* i *dump* programa. Najozbiljniji problem je pomeranje direktorijuma u toku kreiranja rezervne kopije. Naime, ukoliko se direktorijum u toku kreiranja rezervne kopije pomera iz dela sistema koji još nije kopiran, u deo koji je već kopiran može doći do gubitka direktorijuma. Za rešavanje ovog problema koriste se dve metode: zaključavanje i detekcija [3, 4].

Problem konzistencije takođe je moguće prevazići kreiranjem *snapshota* sistema (prim. prev. blic-snimak). *Snapshot* je zamrznuta *read-only* kopija trenutnog stanja fajl sistema.

Problem konzistencije podataka kod *on-line backup* procesa u virtuelnom okruženju je moguće rešiti zamrzavanjem stanja fajla koji opisuje virtualni uređaj. Virtualizacijom uređaja problem *device-based* procesa kreiranja potpune rezervne kopije moguće je svesti na jednostavniji *file-based* pristup. U narednom poglavlju dati su osnovni termini i arhitektura virtualnog okruženja.

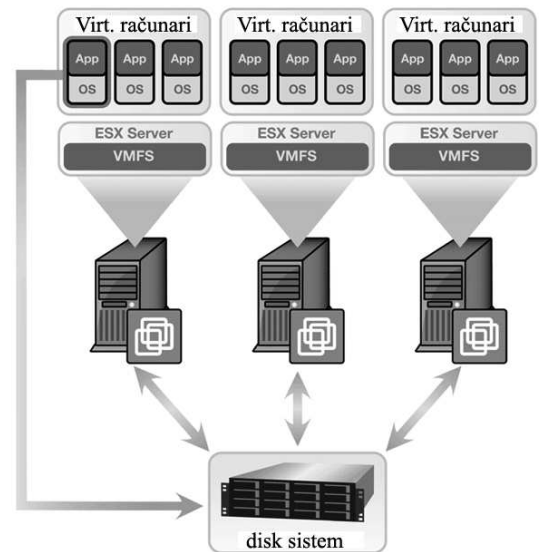
3. VIRTUALIZACIJA SERVISA I SISTEMA

U svojoj ranoj fazi, virtualizacije je 60-tih godina prošlog veka bila poznata pod nazivom *time-sharing* [8]. Tehnika vremenskog deljenja omogućavala je programeru da unosi kod, dok je za istim računarem istovremeno drugi programer debugirao kod. Značajno mesto u evoluciji virtualizacije imaju Atlas i IBM M44/44X računari. Kod Atlas računara prvi put je uveden koncept virtualne memorije, čime je uveden novi sloj abstrakcije nad resursom, što je zajedničko za sve tehnike virtualizacije [5].

Koncept virtualizacije poslednjih godina podrazumeva apstrakciju svih hardverskih resursa računara, tako da je na apstraktnom hardveru moguće pokretati kompletne operativne sisteme. Računar na kome se fizički izvršava virtualizacija naziva se "domaćin", a virtualni servisi "gost" servisi. Pojam "virtualizacija servera" koristi se kada "domaćin" izvršava više "gost" operativnih sistema, pri čemu "gost" operativni sistemi nisu svesni postojanja drugih "gost" sistema, kao ni virtualne platforme koja apstrahuje fizički hardver [5, 6].

Najkritičnija komponenta platforme za virtualizaciju je hipervizor (eng. *hypervisor*). Hipervizor je softver koji omogućava da više virtualnih računara dele isti fizički procesor, memoriju, disk i mrežne resurse, i izvršavaju se na operativnom sistemu domaćina. Primeri dostupnih hipervizora su VMware Workstation/Server, Microsoft Virtual PC/Server i Parallels Workstation. Drugi tip hipervizora su hipervizori koji ne zahtevaju operativni sistem domaćina, pa su im samim tim i performanse bolje [5]. Primer ovakvih hipervizora su VMware ESX, Microsoft Hyper-V i Citrix XenServer. U ovom radu korišćen je

VMware ESX hipervizor, koji je baziran na Linux 2.6 kernelu. Arhitektura sistema sa ESX hipervizorom prikazana je na slici 1.



Slika 1. Arhitektura sistema sa ESX hipervizorom

Iz ugla ESX hipervizora, virtualni server je skup fajlova na fizičkom disku u direktorijumu koji je dodeljen virtualnom računaru. Tipičan listing fajlova direktorijuma u kome su smešteni fajlovi virtualnog računara je prikazan na slici 2.

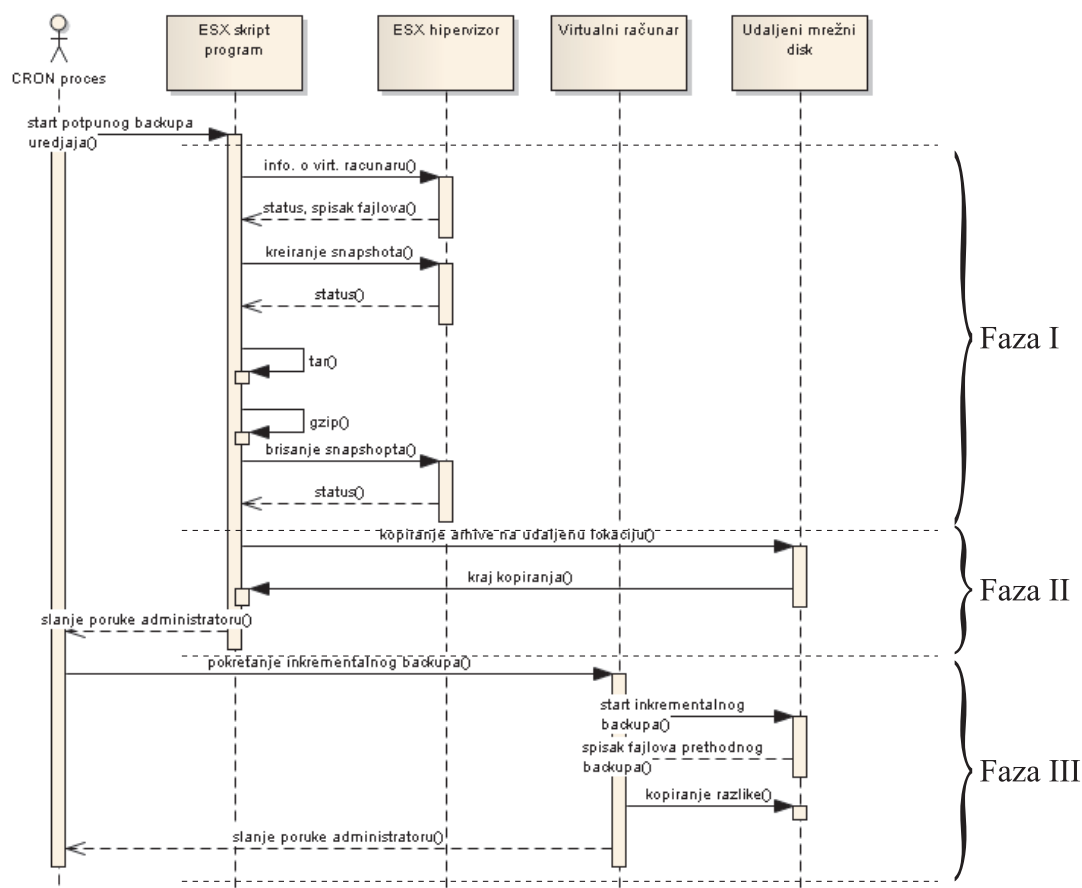
Fajlovi sa ekstenzijom *.vmdk* predstavljaju virtualne diskove virtualnog računara, *.vmsd* služi za pamćenje informacija o *snapshotu* sistema, *nvram* je virtualni BIOS, *vsup* virtualna memorija virtualnog računara, i *vmx* je konfiguracioni fajl virtualnog računara koji povezuje sve delove u celinu [6].

```
total 21G
-rw----- 2.8G Mar 30 21:02 L3public-3b4006ff.vswp
-rw----- 15G Apr 26 19:04 L3public-flat.vmdk
-rw----- 8.5K Apr 26 06:49 L3public.nvram
-rw----- 575 Apr 26 06:49 L3public.vmdk
-rw----- 862 Apr 26 06:03 L3public.vmsd
-rwxr-xr-x 2.6K Apr 26 06:49 L3public.vmx
-rw----- 263 Mar 22 14:38 L3public.vmx
```

Slika 2. Fajlovi koji predstavljaju virtualni računar na ESX hipervizoru

4. KREIRANJE POTPUNE KOPIJE UREĐAJA NA VMWARE ESX HIPERVIZORU

Abstrakcijom fizičkog uređaja na skup fajlova (slika 2) problem kreiranja rezervne kopije uređaja svodi se na problem kreiranja rezervne kopije fajla. ESX hipervizor nudi podršku za kreiranje *shapshota* virtualnog diska kroz VMware skript jezik [7]. Komercijalno, zatvoreno rešenje je VMware Consolidate Backup, koje zahteva postojanje posebnog računara koji upravlja i vrši monitoring *backup* procesa [6].



Slika 3. UML dijagram sekvence izvršenja operacija prilikom kreiranja potpune kopije uređaja na VMWare hipervizoru: Faza I - *snapshot* i kreiranje arhive; Faza II - kopiranje na udaljeni disk; Faza III - inkrementalni backup

Metod predložen u ovom radu ne zahteva postojanje dodatnog računara, već koristi sam ESX hipervizor za kreiranje rezervnih kopija. Na slici 3 dat je UML dijagram sekvence izvršenja operacija predloženog metoda. U kreiranju rezervne kopije virtualnog uređaja na ESX serveru na način predložen na slici 3 učestvuje CRON proces, ESX skript program, hipervizor, virtualni računar i disk za čuvanje kopije, koji je fizički udaljen od sistema, a ceo proces se odvija u tri faze. Linux CRON se koristi za periodično iniciranje *backupa* (slika 3).

U prvoj fazi Linux skript jezikom od hipervizora se zahteva status virtualnog računara, i ukoliko je virtualni uređaj uključen inicira se kreiranje *snapshota* (slika 3). ESX hipervizor nudi podršku za ovo kroz opcije naredbe *vmware-cmd* [7]. Hipervizor kreira *snapshot* tako što prekida upis u *.vmdk* fajl koji predstavlja virtualni disk računara (slika 2), a sve korisničke aktivnosti baferuje u pomoćni fajl koji se kreira u tu svrhu, čime se postiže transparentnost. Skript program koji vodi proces kreiranja rezervne kopije tada uzima sve fajlove, sem pomoćnog *snapshot* fajla, i korišćenjem Linux *tar* programa ove programe pakuje u arhivu i opciono kompresuje Linux *gzip* programom. Ovim se kreira jedna arhiva koja sadrži stanje uređaja u trenutku kreiranja *snapshota*, i koja se u drugoj fazi kopira na udaljenu lokaciju. Pre početka kopiranja skript inicira upis svih baferovanih akcija u originalni *.vmdk* fajl i uklanja *snapshot*.

Proces kreiranja *tar* arhive je zahtevan po pitanju procesorskog vremena i broja obraćanja disku ESX servera i može trajati dugo, pa iz tih razloga nije praktično često vršiti *backup* celog uređaja. Kako se mali broj fajlova menja u toku dana, metod prikazan na slici 3 uvodi treću fazu u okviru koje se relativno češće vrši inkrementalno kopiranje fajlova korisnika sa samog virtualnog računara.

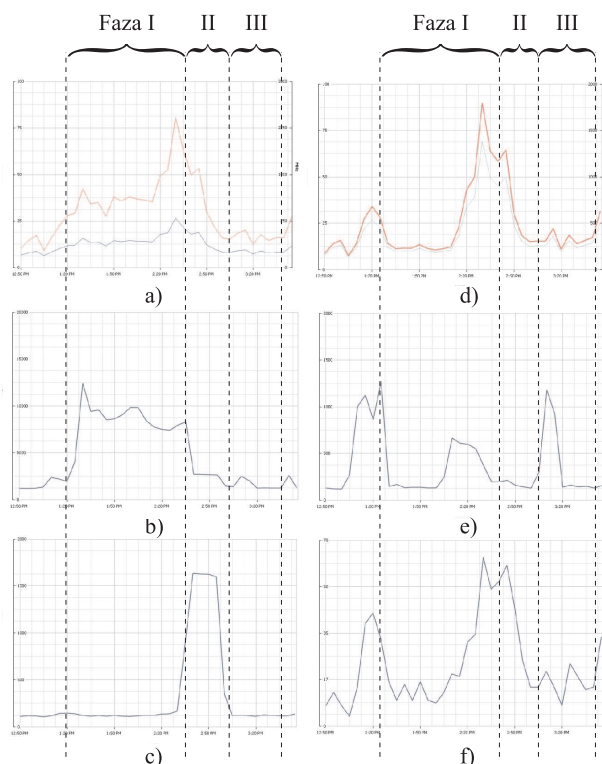
5. IMPLEMENTACIJA SISTEMA

Predloženi sistem je implementiran na dva servera IBM x3550 sa quadcore procesorima, 32GB RAM-a i IBM DS3000 DataStorage mrežnim diskovima od 7.5TB, povezanim na način prikazan na slici 1. Sistemu je pridružen WD MyBook mrežni disk, koji se nalazi fizički daleko od baznog sistema. Na serverima je instaliran ESX hipervizor i implementiran je skript program po dijagramu sekvence datom na slici 3. Kopiranje iz faze II sa slike 3 realizovano je kroz SSH [5]. Potpuni *backup* virtualnog računara inicira se jednom u dva dana, a inkrementalni *backup* dva puta dnevno.

Rezultati opterećenja resursa ESX servera i resursa virtualnog računara prikazani su na slici 4. Na slici 4a, b i c dato je opterećenje procesora, diskova i mreže ESX servera, respektivno, dok je na slici 4d, e i f prikazano opterećenje virtualnog računara. Virtualni računar koji je uzet za primer koristi se, pored ostalog, za izvođenje

laboratorijskih vežbi iz predmeta Algoritmi i programiranje na Elektronskom fakultetu u Nišu, i u vreme snimanja opterećenja datih na slici 4 sistem je imao oko 60 aktivnih korisnika na Moodle platformi.

Sa slike 4 se može uočiti da je u fazi I značajno povećano opterećenje procesora i diskova ESX servera, dok je u fazi II mrežna aktivnost daleko iznad proseka. Za vreme faze I i II virtualni računar ima aktivnost iniciranu akcijama korisnika. U fazi III je aktivnost diskova virtualnog računara povećana usled određivanja skupa fajlova koji su se promenili od poslednjeg *backupa*, a nakon toga javlja se neznatno povećanje mrežne aktivnosti, koje je daleko manje od aktivnosti samih korisnika.



Slika 4. Opterećenje servera u toku kreiranja rezervne kopije: a), b) i c) - CPU, disk i mreža ESX servera, d), e) i f) - CPU, disk i mreža virtualnog računara

U odnosu na *Consolidate Backup* rešenje predloženo od strane VMware-a, rešenje predloženo u ovom radu ne zahteva postojanje nezavisnog računara zaduženog za *backup*, već koristi resurse samog ESX servera. Metod predložen u ovom radu je opravdan u slučajevima kada je odnos opterećenja ESX servera i resursa kojima server raspolaže takav da je moguće nesmetano, u pozadini, kreirati rezervnu kopiju.

6. ZAKLJUČAK

U ovom radu je predložen metod za kreiranje rezervnih kopija virtualnih uređaja, na jednom nivou, i korisničkih podataka na drugom nivou, na takav način da pozadinski proces kreiranja rezervnih kopija ne prekida i ne ometa rad korisnika sistema. Princip transparentnog kreiranja

rezervnih kopija virtualnih servisa ilustrovan je UML dijagramom. Metod je implementiran na VMware ESX hipervizoru VMware skript-jezikom. U radu su dati rezultati opterećenja resursa u procesu kreiranja rezervnih kopija. U odnosu na postojeće rešenje, metod predložen u ovom radu je opravdan kada je odnos opterećenja ESX servera i resursa kojima server raspolaže takav da je moguće nesmetano, u pozadini, kreirati rezervnu kopiju.

LITERATURA

- [1] W. Curtis Preston, "Backup & Recovery", O'Reilly, US, 2007.
- [2] Preston de Guise, "Enterprise Systems Backup and Recovery: A Corporate Insurance Policy", CRC Press, 2009.
- [3] Ann Chervenak, Vivekanand Vellanki, Zachary Kurmas, "Protecting File Systems: A Survey of Backup Techniques", Joint NASA and IEEE Mass Storage Conference, March 1998.
- [4] S. Shumway, "Issues in On-line Backup", In USENIX Proceedings of the 5th Conference on Large Installation Systems Administration, September 1991, pp. 8188.
- [5] John Hoopes, et al., "Virtualization for Security", Elsevier, Inc., 2009.
- [6] Al Muller, Seburn Wilson, "Virtualisation with VMware ESX Server", Syngress Publishing Inc., US, 2005.
- [7] Al Muller, et al., "Scripting VMware", Syngress Publishing Inc., Canada, 2006.
- [8] Christopher Strachey, "Time Sharing in Large, Fast Computers", International Conference on Information Processing, June 1959, pp. 336-341

Abstract –

Recent development of virtualization platforms pushed new concepts, and gave new implementation freedoms in data backup strategies. This paper presents development and implementation of UNIX backup scripts for on-line virtual device-backup, on one level, and incremental backup of user data stored within virtual device on the other level. Online backup of virtual devices will be illustrated by UML sequence diagram. Proposed method will be implemented on VMware platform using VMware script language. Implementation results regarding system resource usage during backup will be given.

Design and Implementation of On-line Data Backup in VMware Virtual Environment

Vladimir Ćirić, Oliver Vojinović