PERFORMANSE EXT3 SISTEMA DATOTEKA: UTICAJ EKSTERNIH JOURNALING TEHNIKA

Borislav Đorđević[1], Nemanja Maček[2], Stanislav Mišković[1], Svetlana Štrbac[2], Vedran Čustović[1]
Institut Mihajlo Pupin-Beograd[1], Viša elektrotehnička škola[2] u Beogradu
e-mail: bora@impcomputers.com

Sadržaj – *Uvod, journaling režimi u ext3 sistemu datoteka pod Linux-om, external jounaling tehnike, metodologija testiranja, test konfiguracija, parametri operativnog sistema, rezultati tesiranja, zaključak*

1. UVOD

Linux je moderan, sofisticiran i moćan operativni sistem. Novije verzije Linux kernela uključuju podršku za rad sa visokoperformansnim journaling sistemima datoteka, poput ext3, ReiserFS, XFS i JFS sistema datoteka. Podrška za ext3 sistem datoteka, čiji je autor Dr Stephen Tweedie, uključena je u većinu distribucija Linux-a, kao što su Red Hat, počev od verzije 7.2, i SuSE, počev od verzije 7.3.

Podrazumevana (default) journaling tehnika je interna koja se realizuje preko log datoteke formirane u istom sistemu datoteka za koji se obavlja journaling. Pored interne journaling tehnike, moguće je realizovati eksternu journaling tehniku, kod koje će se log datoteka realizovati izvan matičnog sistema datoteka. Uređaji za realizaciju eksterne journaling tehnike moraju predstavljati stabilnu memoriju koja ne gubi sadržaj (non-volatile). Tipični takvi uređaji su posebni delovi diska kao efikasno i jeftino rešenje ili NVRAM kao veoma brzo ali dosta skupo rešenje.

Cilj ovog rada je analiza uticaja eksternih journaling tehnika na performanse sistema datoteka, odnosno uporedna analiza performansi ext3 sistema datatoka sa internom i eksternom journaling opcijom. Sistemi datoteka su testirani u identičnom okruženju - svi testovi su obavljeni na identičnom hardveru sa identičnim parametrima kernela, na potpuno istom disku i potpuno istom sistemu datoteka. Ovaj rad se odnosi na analizu performansi NVRAM eksterne journaling tehnike.

2. REŽIMI VOĐENJA DNEVNIKA TRANSAKCIJA U EXT3 FAJL SISTEMU

Prilikom podizanja operativnog sistema proverava se integritet fajl sistema. Gubitak integriteta se najčešće javlja kao posledica nasilnog obaranja sistema, odnosno promena u objektima fajl sistema koje nisu blagovremeno ažurirane u tabeli indeksnih čvorova, i može za posledicu imati gubitak podataka.

Opasnost od gubitka podataka umanjuje se uvođenjem dnevnika transakcija koji prati aktivnosti vezane za promenu meta-data oblasti, odnosno i-node tabele, i objekata fajl sistema. Dnevnik (journal, log) se ažurira pre promene sadržaja objekata i prati relativne promene u fajl sistemu u odnosu na poslednje stabilno stanje. Transakcija se zatvara po obavljenom upisu i može biti ili u potpunosti prihvaćena ili odbijena. U slučaju oštetećenja, izazvanog npr. nepravilnim gašenjem računara, fajl sistem može lako rekonstruisati povratkom na stanje poslednje prihvaćene transakcije.

U ext3 fajl sistemu prisutna su tri režima vođenja dnevnika transakcija: **journal**, **ordered** i **writeback**.

Journal je režim praćenja svih promena u fajl sistemu, kako u meta-data oblasti, tako i u objektima, čime se pouzdanost fajl sistema znatno uvećava na račun performansi. Rednundansa koju ovaj režim rada unosi je velika.

Ordered je režim praćenja promena u meta-data oblasti, pri čemu se promene u objektima fajl sistema upisuju pre ažuriranja i-node tabele. Ovo je podrazumevati režim rada dnevnika, koji garantuje potpunu sinhronizaciju objekata fajl sistema i meta-data oblasti. U odnosu na **journal**, ovaj režim karakteriše manja redundansa i veća brzina rada.

Writeback je režim praćenja promena u meta-data oblasti, pri čemu se i-node tabela može ažurirati pre upisa promena u objekte fajl sistema. Ovo je najbrži režim rada, ali ne garantuje konzistenciju meta-data oblasti, odnosno sinhronizaciju objekata fajl sistema i meta-data oblasti.

3. METODOLOGIJA TESTIRANJA

Postoji nekoliko mogućih scenarija za određivanje performansi fajl sistema. Testiranje se može obaviti pomoću svetski priznatog benchmark softvera, koji simulira različite vrste opterećenja, poput opterećenja Internet Service Provider-a ili NetNews servera. Drugi način uključuje korišćenje specijalnih testova, specijalno dizajniranih u te svrhe, poput testova sekvencijalnog i slučajnog čitanja i pisanja, kreiranja datoteka i simulacije rada u aplikaciji.

Za potrebe ovog rada korišćen je PostMark softver koji simulira opterećenje Internet Mail servera. PostMark kreira veliki inicijalni skup (pool) slučajno generisanih datoteka na bilo kom mestu u fajl sistemu. Nad tim skupom se dalje vrše operacije kreiranja, čitanja, upisa i brisanja datoteka i određuje vreme potrebno za izvršavanje tih operacija. Redosled izvođenja operacija je slučajan čime se dobija na verodostojnosti simulacije. Broj datoteka, opseg njihove veličine i broj transakcija su u potpunosti konfigurabilni, a radi eliminisanja cache efekata preporučuje se kreiranje inicijalnog skupa sa što većim brojem datoteka (bar 10000) i izvršenje što većeg broja transakcija.

4. TEST KONFIGURACIJA

Konfiguraciju za testiranje performansi fajl sistema odlikuju sledeći fundamentalni parametri: matična ploča, vrsta i radni takt procesora, količina i vrsta drugostepene keš

memorije, količina i vrsta operativne (RAM) memorije, tip i model disk kontrolera, tip i model diska.

Peformanse ext3 fajl sistema su testirane na sledećoj konfiguraciji:

Motherboard	Intel Server Board S845WD1-E
Processors	Intel Pentium IV 2.66GHz
L2 onboard cache	512KB
System Bus Speed	533MHz
RAM onboard	512 MB DDR266
System BIOS	Intel WD84510A.86B.0015.P08
Controller	Adaptec 29160 Ultra160 SCSI
Disks	Quantum Atlas V
Operating system	Red Hat Linux 8.0(kernel 2.4.18-14)

Tabela 4.1. Karaktersitike test sistema

Adaptec 29160 je izabran kao reprezentativni kontroler u klasi SCSI kontrolera (non-RAID), dizajniran da opsužuje servere pri nižim i srednjim opterećenjem. Ovaj jednokanalni SCSI kontroler sa 64-bitnom magistalom je idealan za povezivanje Ultra160 SCSI (LVD) diskova, poput diskova iz serije Quantum Atlas-V, kao i drugih internih i eksternih uređaja.

Karakteristike Adaptec 29160 kontrolera i Quantum Atlas-V diska date su u tabelama 4.2 i 4.3 respektivno.

broj SCSI kanala	jedan (single channel)
radno okruženje	serveri pri nižim i srednjim opterećenjem
brzina interfejsa	160 MB/sec
magistrala	64 bit PCI
konektori za interne uređaje	68 pin LVD SCSI
	68 pin Ultra Wide SCSI
	50 pin Eltra SCSI
konektori za eksterne uređaje	68 pin LVD SCSI

Tabela 4.2. Karakteristike Adaptec 29160 kontrolera

average seek time (prosečna brzina pristupa)	6.3ms
full stroke seek (brzina pristupa s kraja na kraj)	15ms
track-to-track seek (brzina pristupa sledećoj stazi)	0.8ms
brzina okretanja ploča	7200 obrtaja u minuti
brzina interfejsa	160 MB/sec
veličina bafera	4 MB

Tabela 4.3. Karakteristike Quantum Atlas-V diskova

5. EKSTERNI JOURNALING I TESTIRANJE

Počevši od verzije 0.9.5, ext3 sistem datoteka podržava koncept eksterne journaling tehnike, kod koje se log datoteka čuva izvan sistema datoteka, obično na posebnoj particiji diska. Druga klasa uređaja koji može da se koristi za eksterni journaling je NVRAM, koji može da se simulira preko memorijskih struktura kao što je RAM disk. U našem slučaju, mi smo NVRAM simulirali preko "trivial RAM disk driver" (/dev/trd by Andrew Tridgell).
Koristili smo NVRAM veličinu u opsegu od 4MB to 128M (power of 2).
Prva naša metoda bila je bazirana na različitim veličinama RAM diska, pri čemu smo ceo RAM disk koristili kao jounrnaling uređaj. Naša prva test metoda, koju smo nazvali "**different sized RAM-disk , whole RAM disk as journal device**", doveo nas je do velike greške simulacije. Sledeći pseudo kod deklariše našu prvu test metodu.

```
for (size = 4M, size <= 128M, size=size*2)
{
insmod trd.o trd_size=size
To create an external journal:
mke2fs -O journal_dev /dev/trd
Whole RAM disk was used as journal devices
First, internal journal must be removed:
tune2fs -O ^has_journal /dev/hda8
Then, we created an external journal on the target devices
mke2fs -J device=/dev/trd /dev/hda8
testing
}
```

Koristeći prvi metodu, došli smo do vrlo zanimljivih, ali neočekivanih rezultata. Pre samim rezultata, očekivali smo sledeće ponašanje. Na malim veličinama NVRAM memorije, očekuju se male performanse, zbog pojave NVRAM "over-flushing", koja znači da se mala log datoteka veoma često prazni. Sa povećanjem NVRAM memorije, očekivali smo da performanse osetno rastu a da se potom dogodi zasićenje, odnosno slučaj kada povećanje NVRAM memorije ne donosi povećanje performansi, zato što se log datoteka veoma malo menja u veličini. Znali smo da sa različitim NVRAM veličinama koje se simuliraju u sistemskoj RAM memoriji, bitno remete uslovi testa, odnosno što je veći RAM disk, to je manji file-caching i obrnuto. Međutim, smatrali smo sa imamo jako puno memorije (512MB) i da na Linux serveru na kome se izvršava jedino PostMark benchmark, uticaj različite veličine RAM diskova neće biti veliki. Ali, prevarili smo se.
Nismo očekivali tako veliki uticaj vliečine za file-caching Na primer u mnogim testovima, performanse za NVRAM veći od 32MB počinju osetno da padaju, što izaziva veliku simulacionu grešku. Rezultati dobijeni na ovaj način nisu bili adekvatni, Nastojali smo da detektujemo i minimalnu i optimalnu NVRAM velićinu, ali jedino smo mogli da detektujemo minimalnu količinu NVRAM memorije.

U tom kontekstu deklarisali smo novu test metodu, koja je eliminisala simulacionu grešku i to se može smatrati najvećim doprinosom ovog rada.
U drugoom metodi, kreirali smo veliki RAM disk (128MB), a journaling datoteku kreirali od dela tog velikog RAM diska (**"one large RAM-disk, part of RAM disk as journal device**"). Na ovaj način, u svim testovima file-caching je isti, zato što je RAM disk fiksne veličine.
Sledeći pseudo kod deklariše našu drugu test metodu.

```
#insmod trd.o trd_size=131072      (For a 128MB device)
To create an external journal using a part of RAM disk:
#mke2fs -O journal_dev /dev/trd size_in_KB

for (size = 4M, size <= 128M, size=size*2)
{
```

To create an external journal:
Part RAM disk was used as journal devices
#mke2fs -O journal_dev /dev/trd size
Then, we created an external journal on the target devices
mke2fs -J device=/dev/trd /dev/hda8
testing
}
Testiranje je vršeno na jednoj od najboljih i često prisutnih distribucija Linux-a, Red Hat sa stabilnom verzijom kernela 2.4.18-14.

Fajl sistemi su kreirani u logičkim particijama na sledeći način:

- boot fajl sistem /dev/sda5, veličine 99MB
- swap particija /dev/sda6, veličine 256MB
- root fajl sistem /dev/sda7, veličine 2.3GB
- test fajl sistem /dev/sda8, veličine 1.3GB

6. REZULTATI TESTIRANJA

Izvršena su tri procene performansi nad različitim skupovima slučajno generisanih datoteka.

1. Test-1 (male i srednje datoteke)

U prvom testu, koji obuhvata testiranje malih i srednjih datoteka, izvršeno je 50000 transakcija nad skupom od 2000 slučajno generisanih datoteka čije se veličine kreću u opsegu 1KB-100KB, što rezultuje čitanjem i pisanjem približno 1.5GB podataka. Ova suma prevazilazi količinu sistemske memorije i generalno eliminiše efekte keširanja diskova. Testovi su trajali u proseku od 7 do 15 minuta.

PostMark konfiguracija:

- set size 1000 100000
- set number 2000
- set transactions 50000

Rezultati testa dati su u tabeli 6.1 i grafički prikazani na slici 6.1.

		RAM sizes					
MB/s	**cls**	**4**	**8**	**16**	**32**	**64**	**128**
read	2.64	1.85	2.1	2.51	3.08	3.35	3.41
write	3.09	2.16	2.46	2.94	3.61	3.92	3.99

Tabela 6.1. *Rezultati prvog testa*

Slika 6.1. Grafički prikaz performansi (prvi test)

U prvom testu, koji obuhvata test malih i srednjih datoteka, detektovano je sledeće ponašanje. Minimalna kolicina NVRAM memorije, koja pobedjuje klasični journaling je 32MB. Detektovana pojava zasicenja na 64MB, performanse veoma malo rastu sa povećanjem veličine NVRAM, 2% sa povećanjem na 128MB. Na 4MB NVRAM slabiji od klasike oko 42% i pojava intenzivnog over-flushing se dešava na 4MB i 8MB sve do 32MB. Svako dupliranje NVRAM do 64 MB solidno povećava performase za oko 10-20%. Iza zasićenja (posle 64MB) NVRAM brži oko 30% u odnosu na klasični journaling. Za ovakvo test oprerećenje dovoljna količina NVRAM memorije je 64MB.

2. Test-2 (ultra male datoteke)

U prvom testu je izvršeno 50000 transakcija nad velikim skupom slučajno generisanih datoteka čije se veličine kreću u opsegu 1bajt-1KB, što rezultuje čitanjem približno 15MB podataka i pisanjem približno 30MB podataka. Ovakva konfiguracija generiše veliki broj zahteva za ažuriranje meta-data oblasti, odnosno i-node tabele. Ovaj test je vrlo intenzivan - ukupna količina podataka za čitanje i upis je znatno manja od količine sistemske memorije, očekuje se veliki uticaj keširanja, ali testovi obiluju ogromnim brojem metadata operacija. Testovi su trajali u proseku od 2.2 do 8.5 minuta.

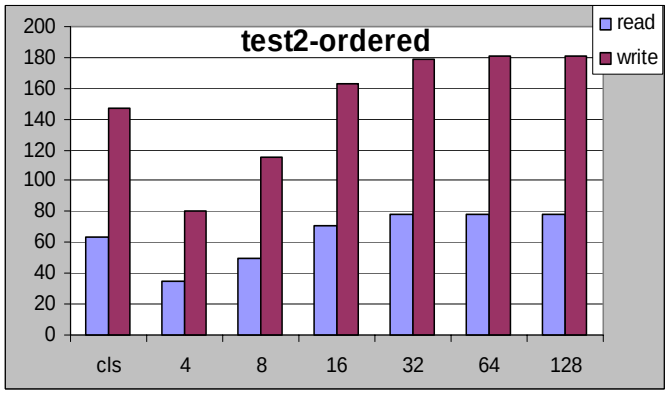
PostMark konfiguracija:

- set size 1 1000
- set number 30000
- set transactions 50000

Rezultati testa dati su u tabeli 6.2 i grafički prikazani na slici 6.2.

		RAM sizes					
KB/s	**cls**	**4**	**8**	**16**	**32**	**64**	**128**
read	63.62	34.75	50.12	70.73	77.84	78.72	78.72
write	146.6	80.05	115.5	162.9	179.3	181.4	181.4

Tabela 6.2. Rezultati drugog testa testa



Slika.6.2. Grafički prikaz performansi (drugi test)

U drugom testu, koji obuhvata test ultra malih datoteka, detektovano je sledeće ponašanje. Minimalna kolicina NVRAM memorije, koja pobedjuje klasični journaling je 16MB. Detektovana pojava zasicenja na 32MB, performanse veoma malo rastu sa povećanjem veličine NVRAM, oko 1% iza 32MB. Na 4MB NVRAM je osetno slabiji od klasike oko 83% i pojava intenzivnog over-flushing se dešava na 4 i 8MB sve do 16MB. Svako dupliranje NVRAM do 32 MB solidno

povećava performase za oko 10-40%. Posle zasićenja (od 32MB pa nadalje), NVRAM je brži oko 24% u odnosu na klasični journaling. Za ovakvo test oprerećenje dovoljna količina NVRAM memorije je 32MB.

3. Test-3 (srednje i veće datoteke)

U trećem testu je izvršeno 50000 transakcija nad skupom od 4000 slučajno generisanih datoteka čija je maksimalna veličina povećana na 300KB, što rezultuje čitanjem i pisanjem približno 5GB written to podataka. Ova suma prevazilazi količinu sistemske memorije i generalno eliminiše efekte keširanja diskova. Testovi su trajali u proseku od 26 do 40 minuta.

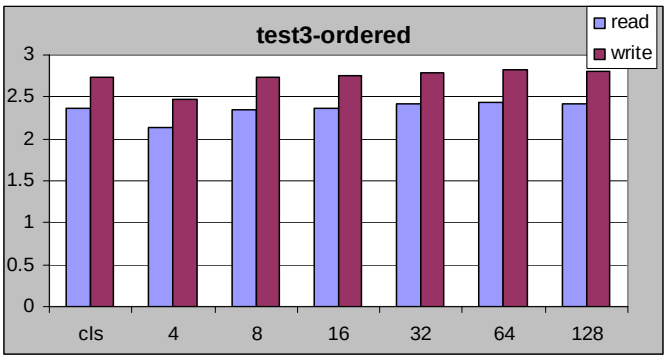
PostMark konfiguracija:

- set size 1000 300000
- set number 4000
- set transactions 50000

Rezultati testa dati su u tabeli 6.3 i grafički prikazani na slici 6.3.

		RAM sizes					
MB/s	**cls**	**4**	**8**	**16**	**32**	**64**	**128**
read	2.36	2.13	2.35	2.37	2.41	2.44	2.41
write	2.74	2.47	2.73	2.75	2.79	2.83	2.8

Tabela 6.3. Rezultati drugog testa testa



Slika.6.3 Grafički prikaz performansi (treći test)

U trećem testu, koji obuhvata test srednjih i većih datoteka, detektovano je sledeće ponašanje. Minimalna kolicina NVRAM memorije, koja pobedjuje klasični journaling je 16MB. Detektovana pojava zasicenja na 32MB, performanse veoma malo rastu sa povećanjem veličine NVRAM, oko 1% iza 32MB. Na 4MB NVRAM je neznatno slabiji od klasike oko 10% i pojava intenzivnog over-flushing se dešava na 4 i 8MB sve do 16MB. Svako dupliranje NVRAM do 32 MB osim sa 4-8M neznatno povećava performase oko 2%. Posle zasićenja (od 32MB pa nadalje), NVRAM je neznatno brži (oko 2-3%) u odnosu na klasični journaling. Za ovakvo test oprerećenje dovoljna količina NVRAM memorije je 32MB.

7. ZAKLJUČAK

Simulirajući NVRAM, neke spektakularne rezultate nismo dobili, na primer faktor povećanja performansi x5 ili x10. NVRAM ubrzava metadata upise (write to log), ali drugi deo journaling procedura se obraca disku, to su log->to->metadata transferi, što može da objasni nedovoljni prinos performasi, korišćenjem NVRAM.

Za podrazumevanu (ordered) journaling opciju karakteristično je sledeće. Minimalna potrebna količina memorije je uglavnom 16MB (32MB za slučaj malih i srednjih datotetka). Zasićenje je detektovano uglavnom na 32MB (64MB za slučaj malih i srednjih datotetka). Sve tranzicije osim onih u zasićenju su beneficijalne, osobito one na malim memorijama (4->8MB, 8->16MB). Za ordered opciju, najveće razlike eksterne i klasične journaling metode su detektovane na za slučaj malih i srednjih datoteke oko 30%, potom u slučaju ultra malih datoteka oko 23%, a najmanje za slučaj srednjih i većih datoteka, svega 2-3%.

Naša NVRAM simulacija, pokazuje da NVRAM za ext3 sistem datoteka ne donosi revolucinarne performanse u odnosu na klasični journaling, a dosta je skup. Mi smo detektovali najveće povećanje performansi od 30%. Za većinu testova, 32-64 MB količina memorije je i minimalna a iistovremeno i optimalna količina. U svakom slučaju, ne preporučujemo NVRAM za ubrzanje journaling tehnike, smatramo da je bolje pokušati external journaling na posebnom delu diska ili probati neki drugi sistem datoteka, reiserFS ili neku drugu journaling opciju.

LITERATURA

[1] Johnson K. M., whitepaper: "*Red Hat's New Journaling File System: ext3*", www.redhat.com/support/wpapers/redhat/ext3/

[2] Tweedie S., "*EXT3, Journaling Filesystem*" July 20, 2000, http://olstrans.sourceforge.net/release/OLS2000-ext3/OLS2000-ext3.html

[3] J. Katcher, "*PostMark: A New File System Benchmark*", Technical Report TR3022. Network Appliance Inc, Oct. 1997.

[4] G. Ganger, Y. Patt, "*Metadata Update Performance in File Systems*", OSDI Conf Proc., pp. 49-60, Monterey, CA, Nov. 1994.

[5] M. Seltzer, G. Ganger, M. McKusick, K. Smith, C. Soules, C. Stein, "Journaling *versus Soft Updates: Asynchronous Meta-data Protection in File Systems*", USENIX Conf. Proc., pp. 71-84, San Diego, CA,June 2000.

Abstract - This paper concentrates on the Linux ext3 filesystem performance comparison problem. Main goal this paper should achieve is analysis of performance impact due to an external journaling approached (NVRAM based), related to internal journaling approached. The performance is measured using by Postmark benchmark software, which emulates Internet mail server environment defined by the authors**.**

PERFORMANCES OF EXT3 FILESYSTEM: IMPACT OF EXTERNAL JOURNALING

Borislav Đorđević, Nemanja Maček, Stanislav Mišković, Svetlana Štrbac, Vedran Čustović