

Delečić Dejan, Lazarević Petar, Milutinović Veljko, Zlatković Slobodan  
Elektrotehnički fakultet  
Bulevar revolucije 73, 11000 Beograd  
tel: 552052(Dejan)

## SIMULACIONA ANALIZA POBOLJŠANOG DISK KEŠ KONTROLERA KOJI RADI U PERIODIMA „TIŠINE”

### SIMULATION ANALYSIS OF AN IMPROVED DISK CACHE CONTROLLER FOR UTILIZATION OF “SILENCE”

**Sadržaj:** U ovom radu prikazani su rezultati simulacione analize poboljšanog disk podsistema baziranog na postojanju disk keša. On pokazuje poboljšanja koja donosi nov i vrlo efikasan algoritam za *prefetch* (dohvatanje unapred) i *rearrangement* (preuređenje) disk keša, koji radi u periodima „tišine” i prati „statički” tok izvršavanja instrukcija, dobijen korišćenjem *trace-scheduling* ili neke druge strategije.

**Abstract:** This work presents the results of a simulation analysis of an improved Reddy/IBM disk subsystem based on disk cache. It shows the benefits of a novel and highly efficient prefetch and rearrangement algorithm, which works in “silence” periods, and follows the “static” instruction address stream, obtained through the utilization of trace-scheduling or similar strategies.

#### 1. Uvod

Dosadašnji razvoj računarske tehnike i poluprovodničkih komponentata u VLSI tehnologiji, učinio je mnoge resurse računara izuzetno moćnim. No, nisu svi delovi računarskog sistema razvijani istom brzinom. Brzina procesora i memorija povećana je mnogo puta, dok je vreme pristupa diskovima, inače podložno ograničenjima mehaničke prirode, smanjeno u mnogo manjoj meri. Postoji i neprestani pritisak za što boljim performansama i što većim kapacitetima za istu ili, čak, nižu cenu. Taj problem je važan zbog postojanja brojnih aplikacija koje karakteriše vrlo intenzivno obraćanje disku (baze podataka, animacije, digitalni video, komponente za podršku virtualne memorije, itd.). Poboljšanje performansi disk podsistema skoro bi linearno poboljšalo performanse celog računarskog sistema za ranije navedene aplikacije. Zato se, danas, o disk podsistemu može govoriti kao o „Ahilovoj peti” radnih računarskih stanica, te je, stoga, velika pažnja posvećena otklanjanju ovog „uskog grla”.

Moderni disk podsistemi sadrže keš memoriju. Ona se nalazi između diska i pozadinske magistrale. Osnovni principi keš memorija, vremenska i prostorna lokalnost, zadovoljeni su sa brojnim parametrima koji karakterišu pristup disku. Na nesreću, većina koncepata disk keš memorija predstavlja, u stvari, samo preslikani koncept procesorskog keša (keša između procesora i operativne memorije). No, takav koncept sigurno nije najoptimalniji način za realizaciju disk keša. Postoje bitne razlike između navedena dva tipa keš memorija, koje se prvenstveno odnose na vreme između dva „naleta” zahteva. To međuvreme (periodi „tišine”), omogućava postojanje novih aktivnosti disk keš kontrolera koje za procesorski keš nemaju nikakvog smisla, dok se kod disk keša postiže veliko poboljšanje performansi.

Osnovne aktivnosti se, kod procesorskog keša, obavljaju prilikom samog obraćanja procesora keš memoriji. Za ovaj keš postoji jedan neprekidan nalet zahteva (“burst of accesses”). Stoga nema vremena koje bi se moglo korisno utrošiti za preuređenje sadržaja keša, između dva sukcesivna zahteva. Kod disk keša, naleti zahteva su praćeni naletima tišine.

Za vreme tih, iz mikro perspektive, dugih perioda tišine moguće je izvršenje, čak i vrlo složenih, algoritama za preuređenje sadržaja disk keša, sa ciljem dobijanja što većih procenata pogodaka u naletima zahteva koji slede. Postoji još nekoliko razlika između procesorskog i disk keša, od kojih su najznačajnije: veličina keša, vreme pristupa, lokalnost, algoritmi za dohvatanje i preuređenje, način realizacije kontrolera, načini adresiranja.

Do sada najbolji disk keš kontroleri (jedan od njih je prezentovao Reddy [11]) koristili su periode tišine za „čišćenje” disk keša (nadalje samo - keša). Sve ostale aktivnosti obavljali su samo prilikom zahteva disku.

Disk keš kontroler, čija je simulaciona analiza prezentovana u ovom radu (pogledati i [3], [4], [5]), koristi u mnogo većoj meri specifičnosti disk keša i pomenute periode tišine, u kojima se, pored čišćenja, vrši i preuređenje keša. Promena sadržaja keša vrši se na osnovu statičkih informacija o toku izvršenja aplikacija i odgovarajućim obraćanjima disku. Te informacije (nadalje - statistike) bi obezbeđivao novi tip prevodilaca (kompajlera) koji bi se rukovodio *trace-scheduling* strategijom analize koda aplikacije (metodologija koja se već koristi za VLIW računare [6]). Na taj način bi se keš popunjavao podacima za koje postoji najveća verovatnoća da će u budućnosti biti traženi. Tako bi se jaz između vremena pristupa operativnoj memoriji i vremena pristupa disku, koji danas iznosi nekoliko redova veličine, bitno smanjio.

## 2. Definicija problema

Cilj simulacione analize jeste upoređenje performansi novog disk keš kontrolera sa najboljim iz otvorene literature [11]. Uslovi simulacija jednako su „povoljni”, odnosno „nepovoljni” za oba kontrolera. Poređenja su vršena za različite veličine keša i veličine bloka u kešu. Takođe, vršeno je poređenje performansi kontrolera za različite vrednosti podudaranja podataka iz ulazne liste simulacije (dakle, stvarno traženih sa diska) sa onima iz statistika, prema kojima se vrši popuna keša u periodima tišine. To je, ustvari, poređenje za različite vrednosti efikasnosti algoritma za generisanje statistika.

Takva poređenja daju dobru osnovu dizajnerima budućih disk podsistema sa keš memorijom da odaberu najoptimalniju veličinu kako bloka u kešu, tako i keša. Takođe, rezultati ovih simulacija mogu dati dobru sliku o dobitcima u performansama sa određenim poboljšanjima kvaliteta statistika.

## 3. Uslovi simulacije

Većina istraživanja se koristi realnim ulaznim listama podataka, dobijenim sa realnih sistema. Ovaj metod, međutim, ima izvesna ograničenja. Između ostalog, realni podaci su vezani za određene uslove pod kojima su dobijeni. Oni često nisu u mogućnosti da iskažu veliku različitost aplikacija. Zbog toga su za ovo istraživanje odabrane sintetičke ulazne liste traženih podataka i statistika. Kao osnova za ovakav pristup korišćen je model detaljno razmotren u [1]. Na taj način je bilo moguće kontrolisati veliki broj parametara, kao što su:

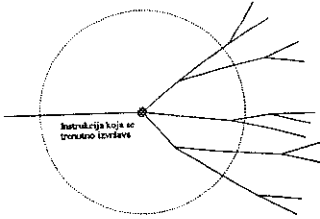
- veličina sekundarne memorije (diska),
- srednja vrednost i disperzija dužine zahteva u klasterima (pretpostavljena je Gauss-ova raspodela),
- srednja vrednost i disperzija vremena između dva zahteva (perioda tišine),
- odnos između zahteva za čitanjem i pisanjem na disk,
- procenat stvarno traženih podataka od onih čije se traženje očekuje.

Ovime je postignuta opštost i fleksibilnost generisanih ulaznih podataka. Tipične vrednosti za navedene parametre odabrane su tako da najbolje pokrivaju široku oblast primene ovakvog kontrolera [4].

Parametri simulacija su veličina keša, veličina bloka u kešu, kao i vreme lociranja podatka na disku. Poslednji parametar je vezan za mehaničke karakteristike diska, dok su dimenzije keša parametri presudni za dobijanje izlaznih rezultata. Uzete su u obzir sve konfiguracije koje imaju smisla u realnim sistemima.

Izlaz iz simulacija je srednje vreme pristupa disku.

Efikasnost simulacione analize veoma zavisi od okruženja na kome se izvodi. Zbog značajnih prednosti odlučeno je da se za simulacije koristi programski paket N.2 [7], [8], [9], [10] na računaru VAX 11-785, pod operativnim sistemom VMS, i na PC računarima, pod operativnim sistemom UNIX. Simulacioni model je realizovan na ISP jeziku za opis hardvera.



Slika 1. Geometrijska predstava strategije dohvatanja podataka.

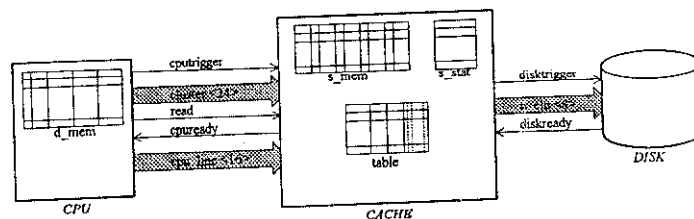
Zadržimo pažnju na osnovnoj razlici ovog kontrolera u odnosu na već postojeće. Za prikaz reorganizacije keša, bazirane na praćenju toka programa koji se izvršava, najbolja je planarna predstava (slika 1.). U tradicionalnim algoritmima za kontrolu keša se ne koriste nikakve informacije o statusu programa koji se izvršava, već se samo zadovoljavaju tekući zahtevi. Stoga se može smatrati da je podatak koji se smešta u keš povezan samo sa instrukcijom koja se trenutno izvršava, a nema nikakve veze sa narednim instrukcijama.

Nova strategija se može predstaviti krugom sa centrom na instrukciji koja se trenutno izvršava (slika 1.). Veličina kruga zavisi od veličine keša i raspoloživog vremena za pripremanje podataka koji u budućnosti mogu biti traženi sa diska. Informacije koje omogućavaju pripremanje sadržaja keša za delove toka aplikacija koji slede jesu statistike. Analitički je pokazano [4] da je za svaku instrukciju obraćanja disku dovoljno postojanje po četiri najverovatnije grupe zahtevanih podataka u statistikama. Za svaku grupu, u statistikama se nalazi broj prvog klastera u grupi, kao i dužina grupe. Pored ovih podataka, statistike mogu sadržati još neke podatke bitne za konfigurisanje i uspešan rad kontrolera. Ovako definisane statistike bi se kreirale za svaku pojedinu aplikaciju, a, takođe, i za delove sistemskog softvera.

#### 4. Struktura simulatora

Simulaciona analiza rađena je pomoću softverskog paketa ENDOT. Izbor ovog programskog paketa načinjen je zbog mnogih njegovih prednosti u odnosu na konkurentne, od kojih se jednostavnost opisa hardvera, kao i velike mogućnosti za skupljanje podataka i statistika iz simulacija, posebno ističu.

Za generisanje statistika i ulaznih podataka za simulacije razvijen je modul čiji je nezavisan rad prethodio glavnim simulacijama. Simulator disk keš podsistema organizovan je u tri glavna modula: CPU, CACHE i DISK (slika 2.).



Slika 2. Struktura simulatora sa označenim vezama između modula i memorijskim elementima u njima.

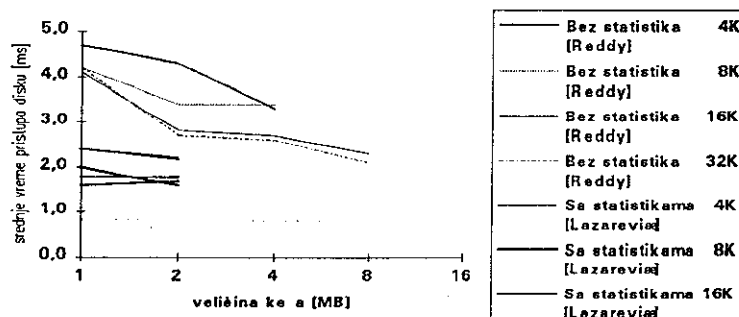
Modul CPU predstavlja deo računarskog sistema koji se završava „ispred“ disk podsistema. Sastoji se samo od jednog procesa koji, na osnovu ulazne liste zahteva, koju je generisao ranije pomenuti modul i koja se nalazi u posebnoj datoteci, generiše nalete zahteva i tišine. Modul DISK realizuje sam disk (*disk drive*) i čine ga dva procesa. Jedan proces, nakon inicijalizacije

njegovog rada od strane odgovarajućih procesa iz modula *CACHE*, posle vremena određenog veličinom jedinice koja se prenosi od keša ka disku ili obrnuto, šalje odgovor procesu koji ga je startovao. Time se simulira, praktično, samo vreme potrebno za obavljanje aktivnosti diska, dok se same aktivnosti ne simuliraju. Drugi proces u modulu omogućava retriggerabilnost diska, resetovanjem inicijalnih vrednosti prvog procesa. Modul *CACHE* opisuje rad disk keš kontrolera. U ovom su modulu realizovani disk keš, algoritmi za pretragu i ažuriranje njegovog sadržaja, algoritmi koji realizuju dohvatanje, preuređenje i izbacivanje blokova podataka iz keša, aktivnosti koje se izvršavaju u periodima tišine, kao i aktivnosti vezane za zadovoljenje zahteva koje generiše modul *CPU*. Tu se, takođe, nalaze i delovi za sakupljanje statistika o radu kontrolera, kao i mehanizmi za zaustavljanje simulacije, nakon što je vreme pristupa disku određeno sa odgovarajućom tačnošću.

Veze postoje između modula *CPU* i *CACHE*, a takođe između *CACHE*-a i *DISK*-a. Komunikacija između modula u sistemu je organizovana po principu zahtev/odgovor. Tipično, u periodima tišine, *CPU* ne generiše zahtev *CACHE*-u, već mu prosleđuje redni broj linije koda aplikacije, koja se trenutno izvršava. Na osnovu tog broja i statistika, koje su povezane sa njima, vrši se preuređenje sadržaja keša. Ukoliko blok podataka iz statistika nije u kešu, *CACHE* generiše zahtev *DISK*-u. Nakon dobijenog odgovora vrši se odgovarajuće ažuriranje LRU lanca i drugih promenljivih potrebnih za rad kontrolera. Postupak se ponavlja sve dok se ne dobije optimalni sadržaj keša ili stigne zahtev od *CPU*-a. Tada se obustavljaju sve aktivnosti vezane za period tišine i aktivira se proces za zadovoljavanje zahteva, koji vrši pretragu keša, generisanje zahteva *DISK*-u, ažuriranja, tj. sve operacije koje su implementirane i u današnjim kontrolerima. Naravno, vrši se i provera uslova završetka simulacije. Nakon zadovoljenog zahteva, tj. dobijenog odgovora od modula *CACHE*, modul *CPU* generiše novi zahtev, ili „generiše“ period tišine.

## 5. Rezultati simulacije

Svi rezultati simulacija dobijeni su uz sledeće uslove i sa sledećim parametrima: Prilikom zahteva za upisom na disk, zahtev se odmah zadovoljava upisom podataka samo u keš (*write-back* strategija), bez utroška vremena. Vreme se jedino gubi prilikom zahteva za čitanjem sa diska, ukoliko se traženi podatak ne nalazi u kešu, a, takođe, i za slučajeve kada je, na zahtev procesora, potrebno novi podatak dovesti u keš koji je sav zaprljan (samo teoretska mogućnost). Iz keša se izbacuju blokovi na osnovu LRU (*least recently used*) algoritma. Blokovi u kešu su veličine od 2 do 16 klastera (4 do 32 KB), dok je veličina keša od 1 do 16 MB. Veličina diska za koje su vršene simulacije je 600 MB. Prosečna veličina naleta zahteva iznosi 8 klastera (16 KB). Prosečno vreme perioda tišine iznosi 1 s.

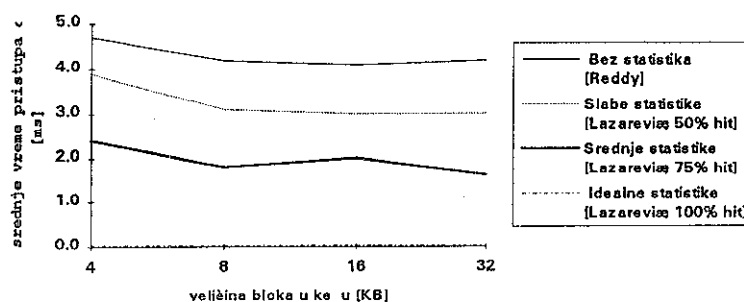


Slika 3. Srednje vreme pristupa disk podsistemu za različite modove rada kontrolera, za različite konfiguracije keša. (Simulacija je još u toku; finalne krive u finalnom radu.)

Grafik sa slike 3. prikazuje uporedne rezultate simulacije kontrolera koji ima mogućnost rada u periodima tišine i praćenja toka izvršavanja aplikacije, i kontrolera bez te mogućnosti. Rezultati su dobijeni kako variranjem veličine keša, tako i variranjem veličine bloka u kešu. Za simulaciju su uzete statistike prosečnog kvaliteta, što konkretno znači da premašuju u 25 % slučajeva.

Značajno je primetiti da su rezultati za oba kontrolera najbolji ukoliko je veličina bloka u kešu jednaka prosečnoj veličini naleta zahteva disku, za svaku veličinu keša. Ovaj rezultat je jako bitan jer su dobijena bitna poboljšanja samo odgovarajućom konfiguracijom keša, uz istu veličinu keša. Uticaj dimenzija postaje značajan i za kontroler koji koristi statistike, u slučaju smanjenja prosečnog vremena između dva naleta zahteva. Zato bi, pri projektovanju disk podsistema, trebalo uključiti i mogućnost konfigurisanja keša prema aplikacijama sa kojima se radi. Parametri te konfiguracije bi se mogli nalaziti u statistikama.

Poboljšanja koja donosi nova strategija iznose u proseku 50 % za navedenu uspešnost statistika. Tako je moguće dobiti jednake, ili čak bolje, performanse disk podsistema za nekoliko puta manju veličinu keša, kada se u isti disk podsistem, umesto najboljeg klasičnog, postavi ovde predstavljeni kontroler. Ovi rezultati, pokazaće se, mogu biti i mnogo bolji.



Slika 4. Srednje vreme pristupa disk podsistemu kontrolera sa statistikama različitog kvaliteta, u odnosu na najbolji klasični kontroler, za keš veličine 1 MB.

Grafik sa slike 4. prikazuje rezultate simulacije kontrolera sa kešom veličine 1 MB. Simulacije su vršene za različite veličine bloka u kešu i različito kvalitetne statistike. Rezultati su prikazani uporedo sa rezultatima kontrolera bez mogućnosti rada u periodima tišine i praćenjem toka izvršavanja aplikacije. Dato je srednje vreme pristupa disku za statistike prosečnog kvaliteta, idealne statistike i statistike koje premašuju u 50 % slučajeva.

Za slučaj idealnih statistika, kod kojih se svi zahtevi koji se, u simulaciji, postavljaju disku nalaze i u statistikama (premašivanje je jednako nuli), vreme pristupa disk podsistemu bilo bi jednako vremenu pristupa memorijskih čipova, koji čine disk keš, tj. bio bi dobijen idealni disk podsistem. To znači da sa variranjem kvaliteta statistika varira i vreme pristupa disku i to tako da je moguće dobiti i rezultate lošije od rezultata kontrolera koji ne koristi statistike, ukoliko bi statistike bile izuzetno loše. Rezultati pokazuju da su, uz korišćenje statistika, parametri keša od manjeg uticaja nego kod najboljeg kontrolera koji primenjuje klasičan koncept rada.

## 6. Zaključak

U ovom radu predstavljeni su rezultati simulacione analize rada disk keš kontrolera koji je nastao kao posledica novog pristupa problemu preuređenja sadržaja disk keša, baziranom na metodologiji praćenja toka izvršavanja aplikacija. Simulacije su pokazale da takva metodologija daje mnogo bolje rezultate od tradicionalne. Implementirani algoritmi obezbeđuju veće

verovatnoće nalaženja željenih podataka u kešu, a samim tim i bitno poboljšanje performansi celog disk podsistema, koje se, tako, približavaju idealnim (performansama operativne memorije). One su limitirane, praktično, samo kvalitetom generisanih statistika. Simulacije takođe pokazuju da se, uz korišćenje prezentovanog kontrolera, iste, ili bolje, performanse mogu ostvariti i za mnogo manje veličine disk keša u odnosu na tradicionalne.

Projektanti disk podsistema, posebno onih koji su namenjeni višeprocensnim računarskim sistemima, mogu imati dosta koristi od ovog rada. To su oblasti u kojima praćenje toka izvršavanja aplikacija može dati izuzetne rezultate za predviđanja koja su od interesa.

Naravno, postoji i niz novih problema koje ovo istraživanje otvara. Jedan od najvažnijih jeste problem generisanja statistika. Od kvaliteta statistika u mnogome zavisi uspešnost rada kontrolera. Sasvim je jasno da se u vreme prevođenja aplikacije ne mogu dobiti optimalne (najbolje!) statistike. Zato je dinamičko ažuriranje tih statistika, od strane dela operativnog sistema (na primer), problem „na tapetu” u bliskoj budućnosti. Tu je i problem praktične realizacije ovakvog kontrolera i njegovih dodatnih veza sa „ostatkom sveta” (veza koje u sadašnjim sistemima ne postoje). Ovom radu neposredno sledi analiza uticaja perioda tišine na performanse disk podsistema. Implementaciona analiza je naredni korak ka konačnoj realizaciji ovakvog disk keš kontrolera.

## 7. Reference

- [1] Archibald, J., Baer, J.L., "Cache Coherence Protocols: Evaluation Using a Multiprocessor Simulation Model," ACM Transactions on Computer Systems, Vol.4, No.4, November 1986, pp. 273-298.
  - [2] Ferrari, D., Computer Systems Performance Evaluation, Prentice-Hall, Englewood Cliffs, N.J., 1984.
  - [3] Lazarević, P., Milutinović, V., "Disk Subsystem Architectures for Fast and Intelligent Data Access in Multimedia Authoring," Technical Report, TI-ETF-RTI-91-009, School of Electrical Engineering, University of Belgrade, Belgrade, Yugoslavia, June 1991.
  - [4] Lazarević, P., Milutinović, V., "Application of Trace Scheduling to Disk Cache Contents on the Silence," Technical Report, TI-ETF-RTI-91-010, School of Electrical Engineering, University of Belgrade, Belgrade, Yugoslavia, June 1991.
  - [5] Lazarević, P., Milutinović, V., Zlatković, S., Deletić, D., "Disk Cache Architecture For the Utilization of the Silence," Technical Report, NCR Corporation, Augsburg, Bavaria, Germany, December 1992.
  - [6] Multiflow Technical Summary, Multiflow Computer, Inc., Branford, Connecticut, U.S.A., June 1987.
  - [7] "N.2 ISP' User's Manual" Document #101, Version 1.14, Zycad Corporation, Menlo Park, California, U.S.A., 8/23/87.
  - [8] "N.2 Ecologist User's Manual" Document #105, Version 1.08, Zycad Corporation, Menlo Park, California, U.S.A., 9/22/87.
  - [9] "N.2 Simulator User's Manual" Document #106, Version 1.21, Zycad Corporation, Menlo Park, California, U.S.A., 7/28/88.
  - [10] "N.2 Tutorial" Document #25, Version 1.01, Zycad Corporation, Menlo Park, California, U.S.A., 6/27/88.
  - [11] Reddy, A.L.N., "A Study of I/O System Organizations," 19th International Symposium on Computer Architecture, Brisbane, Queensland, Australia, May 1992, pp. 308-317.
  - [12] Smith, A.J., "Cache Memories," ACM Computing Surveys, Vol.14, No.3, September 1982, pp. 473-530.
-