Poglavje 1

RAID metodologije kot primer tolerance računalniških sistemov do odpovedi

Pojem *tolerance računalniškega sistema do odpovedi* (angl. *fault tolerant computer*) predpostavlja, da imamo v računalniški sistem vgrajene takšne mehanizme redundance, ki nam omogočajo pravilno delovanje računalniškega sistema tudi v primeru porajanja napak in s tem posledično tudi v primeru odpovedi posameznih komponent sistema. Povedano drugače toleranca računalniških sistemov do odpovedi omogoča procesiranje ob prisotnosti odpovedi, če le njihovo število ni preveliko. Napake in s tem posledično odpovedi posameznih delov sistema so lahko *prehodnega* in *minljivega* (angl. *intermittent*, *transient*) ali pa *trajnega značaja* (angl. *permanent*) [1].

Že v poglavju o teoriji zanesljivosti smo si ogledali zglede TMR in NMR glasovalnih sistemov, ki imajo značilnosti tolerance do odpovedi, v nadaljevanju pričujočega poglavja pa si ogledamo različne konfiguracije RAID diskovnih sistemov (angl. *redundant array of independent disks*). Metodologijo uporabljamo za toleranco trajnih odpovedi posameznih trdih diskov v polju.

1.1 Osnove RAID metodologij

RAID metodologije definirajo načine povezovanja več neodvisnih pomnilnih trdih diskov v sistem trajnega pomnjenja podatkov, ki je lahko po eni plati bolj *zanesljiv* (toleranten do odpovedi posameznih diskov), po drugi plati pa lahko tudi bolj *zmogljiv* z vidika hitrosti izvajanja bralno - pisalnih operacij. Kratica RAID je ob nastanku metodologije koncem osemdesetih let prejšnjega stoletja v neposrednem prevodu pomenila *redundantno polje cenenih diskov* (angl. *re-*

dundant array of inexpensive disks). Z napredkom arhitekture in tehnologije izdelave pomnilnih diskov je prišlo do izničenja predhodne razlike med visokocenovnimi visoko zanesljivimi in nizkocenovnimi manj zanesljivimi diski, zato postane v drugi polovici devetdesetih let prejšnjega stoletja aktualna nova interpretacija kratice, ki predstavlja *redundantno polje neodvisnih diskov* (angl. *redundant array of independent disks*). Slednja je aktualna še danes. Pojem neodvisnosti v intepretaciji kratice izkazuje to, da z vidika gradnje RAID polja proizvajalec, kapaciteta in tip diska niso več pomembni in da v polje lahko integriramo različne trde diske.

V praksi srečamo uporabe različnih vrst RAID konfiguracij ali RAID nivojev (angl. RAID *levels*), ki jih je definiralo združenje SNIA (angl. *Storage Networking Industry Association*) [2]. Opišemo jih v naslednjih razdelkih. Različne konfiguracije slonijo na treh *tehnikah* in sicer na *deljenju podatkov* (angl. *striping*), *zrcaljenju podatkov* (angl. *mirroring*) in *pariteti* (angl. *parity*) [3].

Predpostavimo, da je osnovna entiteta posamezne bralno - pisalne operacije, ki prispe od aplikacije do operacijskega sistema **enota podatkov** in da imamo v RAID polju n diskov. Ob upoštevanju teh predpostavk so razlage predhodno naštetih tehnik povzete po viru [4] opisane v naslednjih razdelkih.

1.1.1 Tehnika deljenja podatkov

Tehnika deljenja (razprševanja) podatkov ali „striping“ tehnika (angl. *striping*), kot jo bomo imenovali v nadaljevanju, izvaja deljenje **enote podatkov** predvidene za pisanje na enako velike, a manjše **segmente podatkov** (angl. *chunk*, *strip*) in v nadaljevanju njihovo paralelno zapisovanje na n razpoložljivih diskovnih enot. Takšno zapisovanje podatkov je zaradi paralelizma procesa pisanja in manjših segmentov podatkov praviloma dosti hitrejše. Postopek branja je obraten in sicer branje v ozadju najprej paralelno prebere množico manjših **segmentov podatkov** iz več diskov v polju, ki jih nato združi v prvotno **enoto podatkov** in nato posreduje operacijskemu sistemu.

Inicializacija RAID sistema s „stripingom“ zahteva vnaprejšnjo definicijo števila sosednjih naslovljivih blokov na posameznih diskovnih enotah, ki bodo predstavljali lokacije za hrambo posameznega **segmenta podatkov** (angl. *strip*) [4]. **Velikost segmenta podatkov** (angl. *strip size*, *strip depth*) izražena v blokih, je največja količina podatkov, ki jo je možno na posamezni disk zapisati ali iz njega prebrati in je na vseh n diskih v polju enaka. Pri tem se seveda poraja vprašanje, kolikšna je idealna velikost segmenta podatkov za specifične bralno - pisalne operacije s posameznega aplikacijskega področja. Množico lokacij segmentov podatkov gledano preko vseh n diskov v polju poimenujemo za **trak podatkov** (angl. *stripe*), posamezne diskovne enote pa za razdeljene na trakove (angl. *striped volume*). **Dolžina traku** je enaka produktu števila diskov in velikosti segmenta podatkov.

Tehnika „stripinga“ omogoča razpršeno hrambo podatkov in doprinaša k pohitritvi izvajanja bralno - pisalnih operacij, saj se tako ena „velika“ bralno pisalna operacija razdeli na množico „manjših“ paralelno izvedenih bralno - pisalnih operacij. „Velikost“ bralno - pisalne operacije je opredeljena z velikostjo

količine podatkov, na katere ta operacija glasi. Zanesljivosti delovanja RAID sistema pomnjenja sama tehnika „stripinga“ ne izboljšuje, temveč jo poslabšuje. V primeru odpovedi posameznega diska v polju tako izgubijo konsistentnost tudi podatki hranjeni na preostalih sicer normalno delujočih $n-1$ diskih.

1.1.2 Tehnika zrcaljenja podatkov

Tehnika zrcaljenja podatkov (angl. *mirroring*) vsako **enoto podatkov** shrani na vsaj dve diskovni enoti od n razpoložljivih, ki je tako hranjena redundantno. V primeru odpovedi posameznega diska v polju so zaradi redundantne hrambe vse podatkovne enote dosegljive s preostalih diskovnih enot. Ob zamenjavi okvarjenega diska RAID logika poskrbi za prenos izgubljenih podatkovnih enot okvarjenega diska s preostalih enot na novo enoto. S tem zadostimo kriteriju odpornosti sistema na posamezne odpovedi, ki smo ga navedli v naslovu pričujočega poglavja. Specifičneje je sistem z zrcaljenjem podatkov odporen vsaj na odpoved enega diska (angl. *single fault tolerant system*).

Tehnika doprinaša predvsem k dvigu zanesljivosti pomnjenja podatkov, v manjši meri pa tudi k pohitritvi bralnih operacij, saj s pomočjo nje lahko dostopamo paralelno do podatkov, ki se v redundantnih kopijah nahajajo na več diskih. V primeru klasične hrambe, kjer so vsi podatki shranjeni na enem disku, paralelno branje ni možno.

1.1.3 Paritetna tehnika

Paritetna tehnika (angl. *parity*) omogoča zvečanje zanesljivosti hrambe podatkov ob prisotnem „stripingu“ brez zrcaljenja in je z vidika potrebne kapacitete pomnilnega prostora za potrebe redundantnih podatkov v primerjavi z zrcaljenjem manj potratna. Osnovna ideja tehnike je, da se pred vsako hrambo **segmenta podatkov** izračunajo njegovi **paritetni podatki**. Po izračunu se slednji shranijo na poseben *paritetni disk* ali pa so shranjeni distribuirano na poljubnem *podatkovnem disku* v RAID polju po „striping“ konceptu. Pri tem še enkrat poudarimo, da so paritetni podatki dodatno breme za hrambo, pri čemer je to redundantno breme manjše od tistega, ki nastopa pri tehniki zrcaljenja. Izračun paritetnih podatkov se običajno izvede na bitnem nivoju s pomočjo logične XOR operacije.

Predpostavimo, da imamo opravka s paritetno tehniko, ki uporablja poseben paritetni disk in $n-1$ podatkovnih diskov. V primeru, da pride do odpovedi paritetnega diska, je sistem hrambe z vidika verodostojnosti hranjenih podatkov še vedno konsistenten. V tem primeru moramo okvarjeni disk zamenjati, RAID logika pa mora ponovno izračunati paritetne podatke glede na hranjene segmente podatkov na podatkovnih diskih in jih shraniti na nov paritetni disk. V primeru, da pride do odpovedi i-tega od $n-1$ podatkovnih diskov, je potrebno okvarjeni disk zamenjati z novim, potem pa zopet na osnovi paritetnih podatkov s paritetnega diska in segmentov podatkov s preostalih $n-2$ podatkovnih diskov izračunati segmente podatkov, ki jih nato zapišemo na i-ti podatkovni

disk. XOR operacija nam zagotavlja na bitnem nivoju dovoljšnjo količino informacije, da je v primerih odpovedi enega diska restavracija podatkov možna (angl. *single fault tolerant system*). Pri tem ne smemo zanemariti potrebnega časa za restavracijo podatkov z diska v odpovedi. V primeru zrcaljenja je ta čas dosti manjši (potrebno je le najprej prebrati, nato pa shraniti predhodno že zrcaljene podatke na ustrezen disk), kot v obeh paritetnih zgledih, saj se v slednjem primerih čas shranjevanja podatkov poveča še za čas izračuna paritetnih podatkov. Slednji je lahko daljši od časa samega branja in shranjevanja na diskovno enoto.

Tehnika paritete doprinaša predvsem k dvigu zanesljivosti pomnjenja podatkov. Njena realizacija je običajno s finančnega vidika cenejša od izvedbe zrcaljenja, ker je količina redundantno pomnjenih podatkov pri paritetni tehniki dosti manjša od količine redundantno pomnjenih podatkov pri zrcaljenju. Po drugi plati izračun paritetnih podatkov predstavlja kar precejšnje procesno breme, ki ga mora izvesti RAID logika ob vsakokratnem pisanju na diskovne enote.

1.1.4 RAID krmilnik

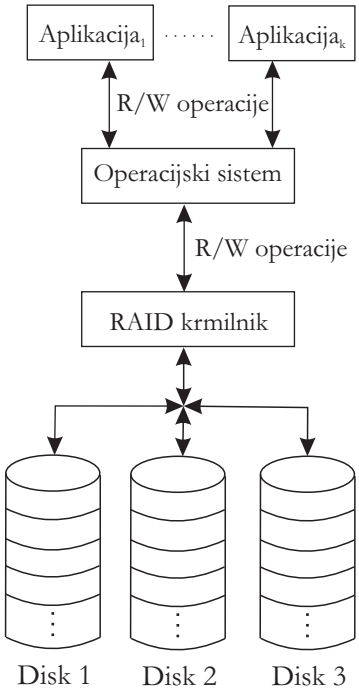
Z logičnega vidika operacijski sistem vidi RAID polje kot en pomnilni medij, kateremu predaja izvajanje bralno - pisalnih operacij [5]. Preslikovanje naslavljanja iz enega logičnega diska na množico fizičnih diskov, zrcaljenja, „striping“ podatkov in paritetne postopke opravljajo *gonilniški programi* (angl. *drivers*) ali pa posebna *namenska strojna oprema*. Ko smo v predhodnjih alineah omenjali „RAID logiko“, smo imeli v mislih prav enega od njiju. V nadaljevanju se bomo na namensko strojno opremo ali na gonilniški program sklicevali s pojmom RAID *krmilnika*, čigar lega v računalniškem sistemu je simbolično prikazana na sliki 1.1.

Gonilniški programi so običajno nameščeni na računalniku, kjer teče tudi programska oprema, ki sproža bralno - pisalne operacije [4]. Tako se moramo zavedati, da z izvajanjem gonilniškega programa opazovani sistem obremenjujemo z dodatnim delovnim bremenom, kar se odraža z zmanjšanjem njegovih razpoložljivih resursov (npr. količine dinamičnega pomnilnika in števila razpoložljivih procesnih ciklov). Prisotnost gonilniškega programa na računalniku, na katerem teče porajanje bralno - pisalnih operacij, tako imenujemo za *invazivno*.

V naslednjih razdelkih opišemo različne RAID konfiguracije. Opisi so večinoma povzeti po viru [4] in posebej poudarimo, da se opisi konfiguracij od vira do vira razlikujejo v podrobnostih izvedbe. V tabeli 1.1 so navedene v nadaljevanju opisane RAID konfiguracije z navedbami uporabljenih tehnik.

1.2 RAID 0

Konfiguracija RAID 0 diskovnega polja temelji na „striping“ tehniki, zanjo pa potrebujemo n diskov v polju ($n \geq 2$). V idealnih razmerah glede na vnaprej definirano velikost *segmenta podatkov* (predpostavimo, da je le ta velikosti k)



Slika 1.1: Umestitev RAID krmilnika kot vmesnika med operacijskim sistemom in poljem treh fizičnih diskov.

in velikost *enote podatkov* (predpostavimo, da je le ta velikosti $k * n$) lahko pridemo do n-kratne pospešitve bralno - pisalnih operacij. Kot nasprotje idealnih razmer lahko navedemo primer, v katerem je enota podatkov enako velika kot vnaprej predvideni segment podatkov, kar pomeni, da pospešitve bralno - pisalnih operacij v tem primeru ne dosežemo.

Kapaciteta RAID 0 pomnilnega prostora je enaka n-kratniku kapacitete najmanjšega diska v polju, kar ponazorimo z izrazom

$$C_{RAID_0} = n * \min(C_1, C_2, ..., C_n), \tag{1.1}$$

pri čemer C_i predstavlja kapaciteto i-tega diska v polju. Metodologija ne doprinaša k zanesljivosti pomnjenja. Odpoved posameznega diska namreč privede neposredno do izgube določenega števila segmentov podatkov ali množice delov trakov, ki niso redundatno hranjeni na preostalih diskih, s tem pa posredno tudi do nekonsistentnosti preostalega dela pomnjenih podatkov v segmentih podatkov na preostalih diskih. Zanesljivost diskovnega polja z n diski v primeru RAID 0 metodologije zapišemo z izrazom

$$R_{RAID_0}(t) = \prod_{i=1}^{n} R_i(t), \tag{1.2}$$

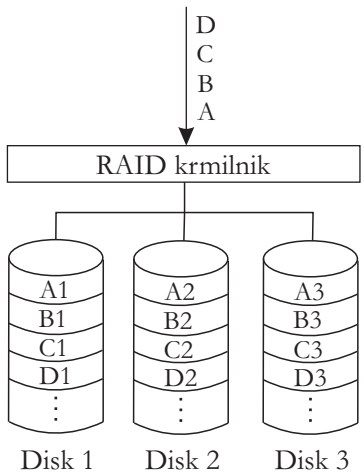
Konfiguracija	„Striping“	Zrcaljenje	Pariteta
RAID 0	x	-	-
RAID 1	-	x	-
RAID 1+0	x	x	-
RAID 0+1	x	x	-
RAID 2	x	-	x(*-HC)
RAID 3	x	-	x
RAID 4	x	-	x
RAID 5	x	-	x
RAID 6	x	-	x

Tabela 1.1: Uporaba tehnik v različnih RAID konfiguracija.

pri čemer je $R_i(t)$ zanesljivost i-tega trdega diska v časovni točki t, intenzivnost odpovedovanja celotnega RAID sistema pa z izrazom

$$\lambda_{sys} = \sum_{i=1}^{n} \lambda_i, \tag{1.3}$$

kjer λ_i predstavlja intenzivnost odpovedovanja i-tega trdega diska. Izraz kaže na zvečanje sistemske intenzivnosti odpovedovanja in s tem posledično na zmanjšanje zanesljivosti sistema kot celote v primerjavi z rešitvijo, ki bi jo predstavljal le en fizični disk. Zapisovanje zaporedja enot podatkov na RAID 0 sistem s tremi diskovnimi enotami je grafično ponazorjeno na sliki 1.2.



Slika 1.2: Zapisovanje zaporedja enot podatkov na tri diske po RAID 0 metodologiji [4].

1.3 RAID 1

Konfiguracija RAID 1 diskovnega polja temelji na tehniki zrcaljenja, zanjo pa potrebujemo n diskov v polju ($n \geq 2$). Vsaka *enota podatkov* se v večini primerov shrani na 2 diskovni enoti, redkeje pa na več diskovnih enot ali na vse diskovne enote.

Predpostavimo, da se vsaka enota podatkov shrani na vse diskovne enote. V tem primeru je kapaciteta tovrstnega polja diskov enaka kapaciteti najmanjšega diska v polju, kar ponazorimo z izrazom

$$C_{RAID_1} = \min(C_1, C_2, ..., C_n). \tag{1.4}$$

Takšno diskovno polje deluje vse dotlej, dokler deluje vsaj ena diskovna enota od n razpoložljivih v polju, saj vsaka diskovna enota hrani vse predhodno zapisane *enote podatkov*.

Najpogostejše RAID 1 konfiguracije vsebujejo dve diskovni enoti ($n = 2$), pri čemer so vse podatkovne enote zrcaljene na obeh diskih. Zanesljivost takšnega diskovnega polja zapišemo z izrazom

$$R_{RAID_1}(t) = 1 - F_{RAID_1}(t) = 1 - \prod_{i=1}^{2}(1 - R_i(t)), \tag{1.5}$$

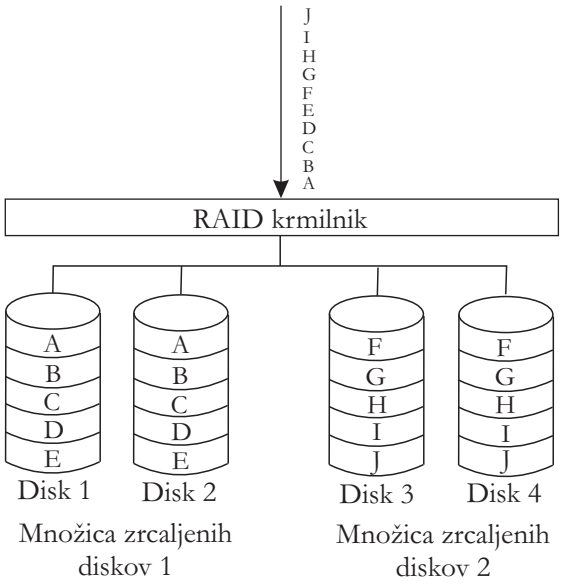
pri čemer je $R_i(t)$ zanesljivost i-tega trdega diska v časovni točki t.

Na sliki 1.3 je predstavljeno zapisovanje zaporedja *enot podatkov* na zgledu štirih diskovnih enot v polju, pri čemer sta po dve diskovni enoti vezani v *zrcalno množico* [4]. V tem primeru se redundantne enote podatkov hranijo le na dveh od štirih razpoložljivih diskovnih enot.

RAID 1 metodologija primarno zvišuje zanesljivost delovanja RAID sistema. Z zmogljivostnega vidika je možno v primeru velikih množic zrcaljenih diskov doseči pospešitev postopka bralnih operacij. Če so podatki zrcaljeni na večje število diskovnih enot, lahko bralne operacije usmerjamo na različne diskovne enote, s tem izvajamo bralne operacije paralelno in tako povečujemo prepustnost podatkov med diskovnimi enotami in operacijskim sistemom. Pri pisalnih operacijah opisana pospešitev ni možna.

1.4 Gnezdene RAID konfiguracije

Gnezdene RAID konfiguracije (angl. *nested* RAID) združujejo performančne prednosti RAID 0 konfiguracij in zanesljivostne prednosti RAID 1 konfiguracij. Dve najpogostejši konfiguraciji sta RAID 1+0 in RAID 0+1. V obeh primerih uporabljamo tehniki „stripinga“ in zrcaljenja, med seboj pa se konfiguraciji razlikujeta le v vrstnem redu uporabe tehnik. Za realizacijo obeh konfiguracij potrebujemo n diskovnih enot, pri čemer je n sodo število.



Slika 1.3: Zapisovanje zaporedja enot podatkov, zrcaljenih na dva diska od štirih razpoložljivih po RAID 1 metodologiji [4].

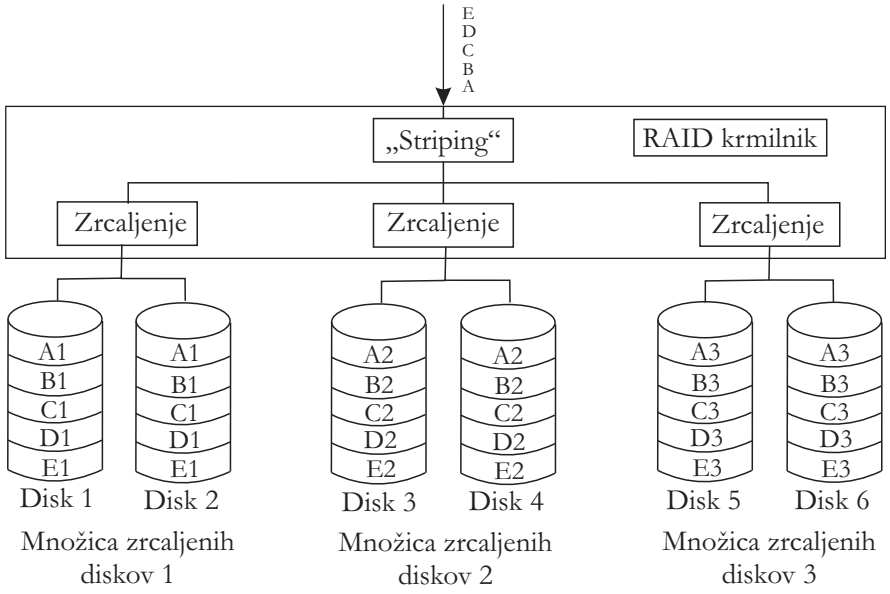
1.4.1 RAID 1+0

RAID 1+0 konfiguracijo označujemo tudi z oznakami RAID 10 ali RAID 1/0. Z vidika zapisovanja podatkov metodologija nad *enotami podatkov* v prvi fazi izvede „striping“, v drugi fazi pa zrcaljenje *segmentov podatkov*. Na sliki 1.4 je predstavljeno zapisovanje zaporedja enot podatkov na zgledu šestih diskovnih enot v polju, pri čemer po dve diskovni enoti tvorita *množico zrcaljenih diskov* [4]. V opisanem primeru lahko odpovedo kar trije diski in to ne ogrozi razpoložljivosti podatkov, če so le vsi trije diski v odpovedi iz različnih množic zrcaljenih diskov.

RAID 1+0 konfiguracija je primerna za aplikacijska okolja, kjer je frekvenca bralno - pisalnih operacij velika in so enote podatkov majhne. Tovrstna aplikacijska okolja najdemo predvsem v OLTP sistemih (angl. *on line transaction processing*) in sistemih za podporo delovanja sistemov podatkovnih baz [4].

1.4.2 RAID 0+1

RAID 0+1 konfiguracijo označujemo tudi z oznakami RAID 01 ali RAID 0/1. Z vidika zapisovanja podatkov metodologija *enote podatkov* v prvi fazi najprej zrcali, v drugi fazi pa se izvede „striping“ enot podatkov na *segmente podatkov*. Na sliki 1.5 je predstavljeno zapisovanje zaporedja enot podatkov na zgledu šestih diskovnih enot, pri čemer so po tri diskovne enote vezane v *množico razpršenih diskov* [4].



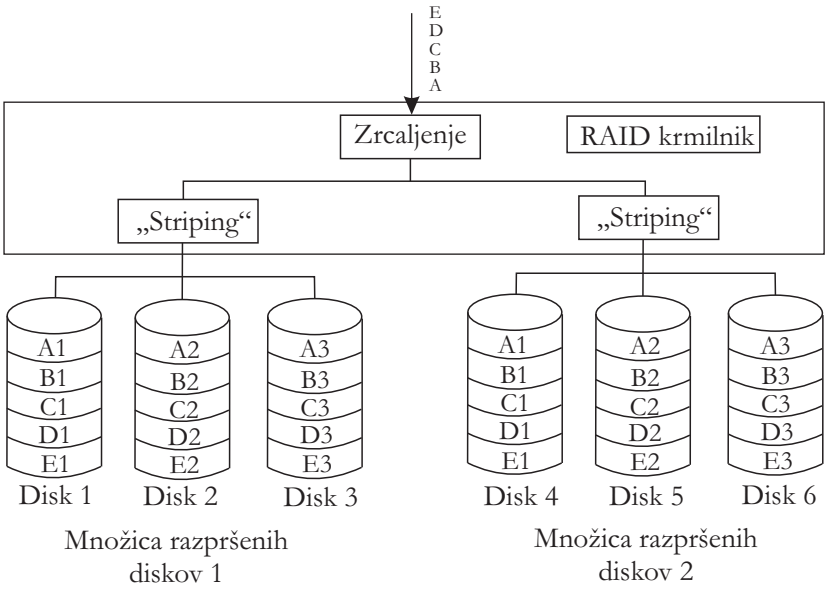
Slika 1.4: Zapisovanje zaporedja enot podatkov, razpršenih in zrcaljenih na dva diska od šestih razpoložljivih po RAID 1+0 metodologiji [4].

1.5 RAID 2

RAID 2 metodologija je izredno redko uporabljana, temelji pa na deljenju podatkov (angl. *striping*) na bitnem nivoju (na posamezen disk se zapiše ali z njega prebere zgolj en bit, ali večje število bitov) in na uporabi Hammingovega korekcijskega koda, ki omogoča tako detekcijo, kot tudi korekcijo napak na posameznih bitih [6].

1.6 RAID 3

Konfiguracija RAID 3 diskovnega polja temelji na tehnikah „stripinga“ in paritete, zanjo pa potrebujemo n diskov. Paritetni podatki se shranjujejo na poseben *paritetni disk*. Odtod v RAID 3 konfiguraciji ločujemo med paritetnim diskom in $n-1$ podatkovnimi diski. RAID 3 konfiguracija omogoča le bralno - pisalne operacije, ki glasijo na celotne trakove (angl. *stripe*). Slednje pomeni, da nimamo možnosti izvajanja operacij, ki glasijo zgolj na posamezne segmente podatkov (angl. *strip*) ali manjše dele trakov. Kakršnakoli bralno - pisalna operacija se tako venomer izvede na nivoju celotnega polja n diskov [4]. Omenjeno značilnost poimenujemo za *odvisno dostopnost*. Na sliki 1.6 je predstavljeno zapisovanje zaporedja enot podatkov na zgledu štirih podatkovnih diskov in enega paritetnega diska. Za podani primer lahko ocenimo, da je razmerje med celotno



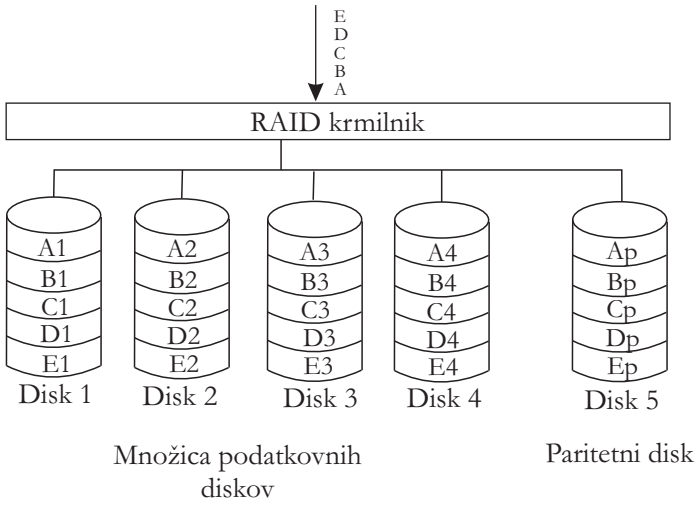
Slika 1.5: Zapisovanje zaporedja enot podatkov, zrcaljenih in razpršenih na tri diske od šestih razpoložljivih po RAID 0+1 metodologiji [4].

diskovno kapaciteto in kapaciteto podatkovnih diskov 1,25, v primeru, da so vsi diski enakih kapacitet. Omenjeni koeficient izraža stopnjo redundance in je manjši od razmerja, ki ga dosegamo pri zrcaljenju in je po vrednosti najmanj 2.

RAID 3 konfiguracije so zaradi možnosti pospešitve bralno - pisalnih operacij primerne za aplikacijska področja, kjer prihaja do branj velikih količin podatkov, ki so shranjeni na zaporednih lokacijah. Mednje sodijo multimedijski podatki (npr. branje za potrebe video predvajanja (angl. *video streaming*)) in varnostne kopije podatkov (angl. *backup data*).

1.7 RAID 4

Konfiguracija RAID 4 diskovnega polja je podobna konfiguraciji RAID 3 polja. Temelji na „stripingu“, pariteti ter n diskovnih enotah v polju, pri čemer je en disk namenjen paritetnim podatkom, ostali pa podatkovnim. Enota bralno - pisalne operacije ni več celoten trak podatkov (angl. *stripe*), temveč je ta enota lahko manjša. Tako lahko bralno pisalne - operacije glasijo tudi na posamezne *segmente podatkov*, kar poimenujemo za *neodvisno dostopnost do podatkov* (angl. *independent accessability*).



Slika 1.6: Zapisovanje zaporedja enot podatkov, razpršenih in opremljenih s paritetnimi podatki na štiri podatkovne diske in en paritetni disk po RAID 3 metodologiji [4].

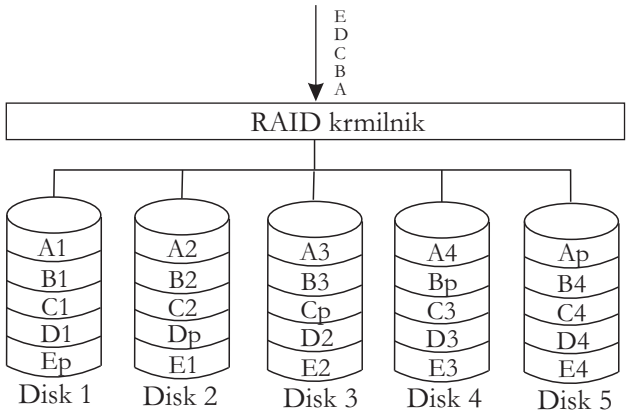
1.8 RAID 5

Konfiguracija RAID 5 diskovnega polja temelji na tehniki „stripinga“, neodvisni dostopnosti do podatkov (možnosti dostopa do posameznih segmentov podatkov) in tehniki paritete. Razlika med RAID 4 konfiguracijo je v mestu hrambe paritetnih podatkov, ki v primeru RAID 5 ni več centralizirana, temveč *distribuirana* po celotnem diskovnem polju. Za realizacijo RAID 5 diskovnega polja potrebujemo najmanj tri diskovne enote ($n \geq 3$). Na sliki 1.7 je predstavljeno zapisovanje zaporedja enot podatkov na zgledu petih diskovnih enot.

RAID 5 sistem je toleranten do odpovedi enega diska v polju. Z uporabo RAID 5 metodologije se izognemo ozkemu grlu (angl. *bottleneck*), ki se poraja na paritetnem disku pri RAID 4 metodologiji. RAID 5 konfiguracijo uporabljamo za aplikacije, kjer prevladujejo bralne operacije. Primera slednjih sta npr. podatkovno rudarjenje (angl. *data mining*) in RDBMS sistemi (angl. *relational database management system*).

1.9 RAID 6

Konfiguracija RAID 6 polja je podobna konfiguraciji RAID 5 polja (torej temelji na „stripingu“, neodvisni dostopnosti do podatkov ter tehniki paritete), pri čemer je dodan drugi segment paritetnih podatkov (angl. *dual parity*). Realizacija RAID 6 polja zahteva minimalno 4 diskovne enote ($n \geq 4$). Konfiguracija je zanimiva, ker je ob redundanci paritetnih podatkov sistem možno restavrirati

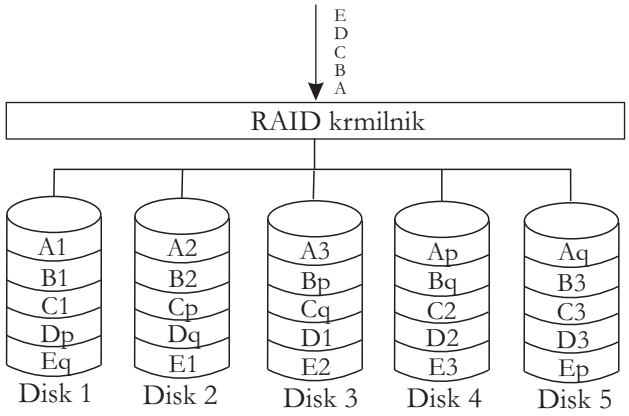


Slika 1.7: Zapisovanje zaporedja enot podatkov, razpršenih in opremljenih s paritetnimi podatki na pet diskov po RAID 5 metodologiji [4].

tudi ob odpovedi dveh diskovnih enot, s čimer se zviša zanesljivost sistema, ki bi jo v tem primeru lahko ponazorili s terminom „$(n-2)$ out of n sistema“ ali matematičnim izrazom

$$R_{sys}(t) = \sum_{r=n-2}^{n} \left(\begin{array}{c} n \\ r \end{array} \right) R(t)^r * (1 - R(t))^{n-r}. \tag{1.6}$$

Na sliki 1.8 je predstavljeno zapisovanje zaporedja enot podatkov na zgledu petih diskovnih enot.



Slika 1.8: Zapisovanje zaporedja enot podatkov, razpršenih in opremljenih z dualnimi paritetnimi podatki na pet diskov po RAID 6 metodologiji [4].

Restavracija podatkov ob odpovedi enega od diskov je pri RAID 6 počasnejša

v primerjavi z RAID 5 konfiguracijo, ker je za restavracijo potrebno izvesti več bralnih in računskih operacij, zaradi večje količine redundantnih podatkov.

1.10 Dodatne metode za zvišanje zanesljivosti RAID sistemov

V vseh naštetih RAID konfiguracijah lahko v diskovno polje dodamo tudi dodatni *rezervni disk* (angl. *hot spare disk*), ki lahko v primeru odpovedi enega od preostalih diskov služi kot začasna lokacija, na katero se rekonstruira podatke z diska v odpovedi [7]. Logika RAID krmilnika v primeru uporabe tehnike zrcaljenja izvede kopiranje podatkov na rezervni disk, v primeru uporabe tehnike paritete pa izračun manjkajajočih podatkov in njihov zapis na rezervni disk. Če hočemo funkcijo rezervnega diska na daljši rok ohraniti, moramo izvesti v nadaljevanju tudi zamenjavo diska v odpovedi. Rekonstrukcija podatkov iz diska v odpovedi se lahko izvede avtomatizirano, ali pa je sprožena interaktivno.

Ena od možnosti, ki nam jo ponujajo RAID sistemi, je tudi menjava posameznega diska v polju v času delovanja RAID diskovnega polja (angl. *hot swaping*). V obeh opisanih primerih ne pride do izpada storitev RAID diskovnega polja, kar pomeni, da sistem deluje neprestano ob prisotnosti odpovedi, če le število slednjih ni preveliko.

1.11 Breme RAID krmilnika

Pri izbiri RAID konfiguracije moramo biti pozorni na doseganje zmogljivostnih in zanesljivostnih ciljev aplikacijskega okolja, v katero bo umeščeno diskovno polje. Pri tem se moramo zavedati procesnih bremen, ki nastajajo v ozadju bralno - pisalnih operacij, ki jih izvaja RAID krmilnik. Slednje lahko drastično upočasnijo delovanje sistema. Procesne breme, ki nastaja v domeni RAID krmilnika, ponazorimo z dvema zgledoma.

Predpostavimo, da imamo opravka z RAID 5 konfiguracijo, predstavljeno na sliki 1.7. Kot smo že povedali, omenjena konfiguracija omogoča neodvisnost dostopa (branja ali pisanja) segmenta podatkov. Predpostavimo, da pride do ponovnega zapisa podatkov v sklopu podatkovnega segmenta $A4$ [4]. V ta namen je potrebno poleg samega zapisovanja segmenta $A4$ ustrezno ažurirati tudi paritetni segment Ap po izrazu

$$Ap_{new} = A1_{old} + A2_{old} + A3_{old} + A4_{new}. \tag{1.7}$$

Navedeni izraz lahko skrajšamo, glede na to, da imamo že predhodno izračunani izraz

$$Ap_{old} = A1_{old} + A2_{old} + A3_{old} + A4_{old} \tag{1.8}$$

in sicer iz slednjega izraza v predhodni izraz vstavimo razliko $Ap_{old} - A4_{old}$ in dobimo izraz

$$Ap_{new} = Ap_{old} - A4_{old} + A4_{new}. \tag{1.9}$$

Iz izrazov je razvidno, da je potrebno pri zapisovanju A4 tako izvesti dve branji segmentov podatkov ($A4_{old}$ in Ap_{old}), izračunati novo vrednost Ap_{new} in na koncu zapisati dva nova segmenta podatkov ($A4_{new}$ in Ap_{new}). V tem primeru z vidika bralno - pisalnih operacij govorimo o kazenskem pribitku (angl. *penalty*) 4 operacij.

V primeru RAID 6 konfiguracije s slike 1.8 bi bil ta pribitek še večji, saj so redundantni podatki hranjeni dvakratno v dualni obliki. Kazenski pribitek bralno pisalnih operacij v tem primeru bi bil 6 operacij.

1.12 Povzetek RAID metodologij

Osnovni kriteriji za vzpostavitev in izbiro konfiguracije RAID sistema so željena zanesljivost, željena zmogljivost in cena.

RAID sistemi so v zadnjem desetletju začeli prodirati kot dokaj zanesljiva rešitev hrambe podatkov tudi k domačim končnim uporabnikom. V predhodnih razdelkih razložene različne metodologije predstavljajo samo osnovne koncepte RAID arhitektur, ki se s hitrim širjenjem do mest končnih uporabnikov tudi hitro nadgrajujejo in oblikujejo specifičneje glede na potrebe uporabnikov in novo nastajajočih storitev.

Ključni problem uporabe RAID metodologij z zanesljivostnega vidika se nahaja v fizično centralizirani lokacijski postavitvi RAID sistemov. Slednje pomeni, da je sistem v redundančnih izvedenkah (RAID 1, RAID 10 itd.) odporen na odpovedi posameznih diskov, je pa ranljiv z vidika zunanjih dejavnikov kot so požar, poplave, potres itd., ki ogrozijo sistem kot celoto, kar vodi do odpovedi vseh komponent sistema. Zato je ključnega pomena, da RAID sisteme smatramo le kot relativno zanesljiv, a neidealen sistem za hrambo podatkov, ki pa mora v primeru hrambe pomembnejših podatkov še vedno biti dopolnjen z fizično oddaljenim sistemom za arhiviranje njihovih varnostnih kopij v skladu s predvidenim načrtom varnostnega kopiranja podatkov (angl. *backup plan*).

Literatura

[1] "D. A. Rennels: Fault - tolerant computing." `http://web.cs.ucla.edu/~rennels/article98.pdf`, April 2018.

[2] "Storage Networking Industry Association." `https://www.snia.org/`, April 2018.

[3] M. L. Shooman, *Reliability of computer systems and networks: fault tolerance, analysis, and design*. J.Wiley and Sons, 2002.

[4] "Data protection: RAID," in *Information storage and management* (S. Gnanasundaram and A. Shrivastava, eds.), ch. 3, pp. 51–68, John Wiley & Sons, Inc., 2012.

[5] W. Stallings, *Computer organization and architecture: Designing for performance*. Prentice Hall Inc., 1996.

[6] "RAID levels." `https://en.wikipedia.org/wiki/Standard_RAID_levels/`, Maj 2018.

[7] J. L. Hennessy and D. A. Patterson, *Computer architecture: A quantitative approach*. Morgan Kaufmann Publishers Inc., 2003.