Kratka priča o NAS i SAN sustavima (i malo više)

Autor : Hrvoje Horvat

Ova knjiga je napisana unutar inicijative *Open Source Osijek*: https://www.opensource-osijek.org

Inicijativa *Open Source Osijek* je član udruge *Osijek Software City*: http://softwarecity.hr/

Mjesto i godina izdavanja: Osijek, 2017

Verzija publikacije : 1.0

Naklada : Vlastita naklada

DokuWiki URL (HTML): https://www.opensource-osijek.org/dokuwiki/wiki:knjige:kratka-prica-o-nas-i-san-sustavima
ISBN 978-953-59438-3-9 (HTML-online)

URL publikacije (PDF): https://www.opensource-osijek.org/knjige/Kratka priča o NAS i SAN sustavima (i malo više).pdf
ISBN 978-953-59438-4-6 (PDF)

Sadržaj

Predgovor

Prvo ćemo naučiti ponešto o diskovima, RAID kontrolerima i RAID poljima diskova. Vidjeti ćete koje su razlike između raznih RAID polja diskova ali i naučiti nešto o particijama i datotečnim sustavima. U daljem tekstu upoznati ćemo se s NAS (Network Attached Storage) i SAN (Storage Area Network) sustavima za mrežnu pohranu podataka. Naučiti ćemo što je dijeljenje datoteka preko mreže, i kako to sve radi. Nadalje vidjeti ćemo što su NAS i SAN sustavi i u čemu su razlike među njima. Dobiti ćete i smjernice što bi još trebali naučiti ili na što paziti ako sami želite postaviti svoj NAS ili SAN sustav. I na kraju, vidjeti ćemo u kojem smjeru se razvijaju ove tehnologije. Dodatno ćemo se upoznati s jednom od naprednih klasterskih tehnologija za pohranu i dijeljenje podataka preko mreže, namijenjene najvećim sustavima, s potrebama za korištenjem velike količine podataka - veličine nekoliko PB (*Petabyte* je 1 000 TB) ili više.

“ **Budi promjena koju želiš vidjeti u svijetu !.** ”

Mohandas Karamchand Gandhi (Mahatma Gandhi)

O Autoru

Hrvoje Horvat je rođen 1975 godine u Osijeku, gdje je završio Prvu tehničku školu, smjer elektronika i automatika. Od prvog druženja s računalom, sredinom 80-tih (ZX Spectrum) i prvih igara a potom prelaskom na Atari ST i kasnije na PC arhitekturu, odabir budućeg zanimanja mu je postao vrlo jasan. Prvo umrežavanje računala s kolegama, u naselju, tijekom 1994, doprinijelo je novim smjerovima osobnog rada i razvoja. Uslijedili su razni poslovi održavanja računala i mreže te se 2000. godine zapošljava u tvrtki ***Siemens d.o.o*** u kojoj i danas radi.

Prvih nekoliko godina radi kao sistem administrator, potom sistem i mrežni inženjer, na korporativnoj IT infrastrukturi tvrtke, te na nekoliko većih infrastrukturnih projekata. Pohađa razne specijalizacije, kako vanjske tako i unutar tvrtke. Zadnjih desetak godina radi kao razvojni inženjer, na projektima koji obuhvaćaju : visoko dostupne i redundantne sustave, AAA i PCRF servise, Virtualizaciju i NAS/SAN sustave te napredne mrežne servise i protokole.

Suosnivač je udruge za razvoj IT sustava *Udruga01*, unutar koje se pokreću deseci projekata razvoja i testiranja raznih mrežnih uređaja, protokola i tehnologija, u cilju učenja i stjecanja novih znanja. Suosnivač je inicijative „Open Source Osijek“, koja potiče i promovira razvoj i upotrebu sustava otvorenog koda, u kojoj i aktivno objavljuje stručne članke. Član je udruge *Osijek Software City* u kojoj djeluje kao aktivni član i predavač.

RAID i pažnja pri odabiru diskova

Prije nego se upoznamo s **NAS** ili **SAN** sustavima, moramo se upoznati s **RAID** tehnologijama te s osnovnom podjelom tvrdih diskova. Naime bez ovih osnovnih znanja, nemoguće je kvalitetno krenuti dalje, s usvajanjem novih znanja iz područja NAS i SAN sustava. Stoga sve što slijedi u ovom poglavlju naučite (kao tablicu množenja).

Krenimo od diskova

Diskovi se u grubo dijele prema namjeni. U svakom slučaju želimo diskove koji su pouzdani ali i koji su proizvedeni za Tzv. Serversku (poslužiteljsku) namjenu u kombinaciji s RAID kontrolerima.

Budite svjesni činjenice da postoje i **serverski** diskovi koji nisu dizajnirani odnosno optimizirani za rad s RAID kontrolerima.

Diskove prema namjeni možemo podijeliti u dvije grupe:

- Diskovi za profesionalnu upotrebu - za poslužitelje ili namjenu 24/7 (365 dana godišnje)
- Diskovi za *kućnu* upotrebu

Ovdje se prvenstveno radi o pouzdanosti ali često i o brzini rada tvrdog diska.

Važan parametar svakog diska, vezan za brzinu rada je i broj okretaja samog diska, te često imamo slijedeće rotacijske brzine:

- 5400 o/min (Engl. *rpm* (Rounds Per Minute))
- 7200 o/min
- 10000 o/min
- 15000 o/min

Što su veće rotacijske brzine ploča diskova (okretaja/minuti) to je veće i zagrijavanje ali i brzina prijenosa podataka

Što su veće rotacijske brzine ploča diskova (okretaja/minuti) to je veće i zagrijavanje ali i brzina prijenosa podataka

Usporedit ćemo tri tvrda diska tvrtke **Western Digital**, oba sa *SATA* sučeljem ali jedan za kućnu upotrebu (WD Black) i dva za profesionalnu primjenu (WD RE) i (WD SE):

Funkcionalnost / Model diska	**WD Black (FZEX ili FAEX)**	**WD RE (FYYZ ili FSYZ)**	**WD SE (F9YZ)**
MTBF (sati)	Nedefinirano	1 200 000	1 000 000 - 1 200 000
Load unload Cycles	300 000	600 000	300 000
Broj nepopravljivih grešaka prema bitovima pročitanih podataka	manje od 1 na 10^{14} bitova	manje od 1 na 10^{15} bitova manje od 1 na 10^{16} za nove generacije (RE4)	manje od 1 na 10^{15} bitova
AFR %	Nedefinirano	0.73	Nedefinirano
TLER	NE	DA	DA
Zaštita od vibracija	Stable Trac, VCT	Stable Trac, VCT, RAAF	Stable Trac, Enhanced RAFF

Iz tablice je vidljivo sljedeće:

MTBF ili engl “Mean time between failures” je predviđeno vrijeme rada (u satima) diska prije kvarova ili grešaka u radu. Odnosno predviđen radni vijek samog diska.

Load unload Cycles je broj pokretanja diska. Naime nakon svakog paljenja ili gašenja računala, disk se pokreće odnosno servo motor “zavrti” ploče (i vrti ih konstantnom brzinom). Ovdje se praktično radi o tome koliko puta se disk može stopirati i startati. Primjerice kada disk ide u “Sleep”, motor se zaustavlja i disk se gasi a kod izlaska iz “Sleep” faze, disk se ponovno uključuje i “zavrti”.

Broj nepopravljivih grešaka prema bitovima pročitanih podataka je vrsta greške na koju će se naići s vremenom i koja se kao što i naziv kaže - neće moći popraviti. Naime svaki tvrdi disk s vremena na vrijeme nailazi na greške te ih popravlja u radu, da mi toga nismo niti svjesni. Ali također može doći do pogrešaka koje elektronika diska ne može popraviti - to su upravo ove greške koje su ovdje navedene. Vidljivo je da profesionalne serije diskova imaju 10 (deset) puta veću pouzdanost od "običnih" diskova.

AFR ili engl. "Annualized failure rate" daje nam vjerojatnost da će se disk pokvariti u toku cijelogodišnje stalne upotrebe. Vjerojatnost od 0.73 % za disk znači vjerojatnost da se disk pokvari unutar jedna cijele godine rada (tj. 8760 sati konstantnog rada).

TLER ili engl. *Time Limited Error Recovery*. Ovu opciju drugi proizvođači poput Samsung/Hitachi, nazivaju i CCTL (Command Completion Time Limit). Prvo se vratimo na osnove - kao što sam rekao svaki tvrdi disk u pozadini stalno popravlja greške na koje povremeno nailazi. Kod običnih diskova to nije problem - ponekad možemo primijetiti da disk nešto radi a da zapravo ne zapisuje ili čita neke nama vidljive podatke na disk. U tim trenucima naš disk vjerojatno u pozadini sam popravlja određene greške, i to može potrajati neko vrijeme (i nekoliko minuta). U normalnom radu nam to nije važno ali ako se disk koristi unutar nekog RAID kontrolera to postaje problem. Naime RAID kontroler očekuje da ako neki disk ima greške a koje sam disk ne može popraviti u roku od nekoliko sekundi do max. 8. sekundi, da disk prestane s popravljanjem. Dakle ako disk ne popravi grešku za max. 8 sekundi, on mora prestati s popravljanjem i prijaviti grešku RAID kontroleru, koji će tada taj disk proglasiti neispravnim. Ako se ovo ne bi dogodilo cijelo RAID polje bi se zaustavilo s radom dok se problematični disk ne popravi (ako se uopće uspije popraviti). Dakle ovo je vrlo važna funkcionalnost za diskove koje koristimo u RAID poljima.

Zaštita od vibracija - *Standardni* diskovi imaju klasične mehanizme zaštite od vibracija dok oni za profesionalnu primjenu moraju imati napredne mehanizme zaštite. *RAFF* tehnologija je jedna od njih (engl. *Rotary Acceleration Feed Forward*). *RAFF* tehnologija pomaže očuvanju performansi diska ali i povećanju njegovog radnog vijeka. Naime ovdje je naglasak na stabilnost rada diska, koju mogu narušiti vibracije drugih (susjednih) diskova, ventilatora ili slično. Zamislimo poslužitelj koji ima više od dva diska od kojih svaki stvara svoje vibracije i prenosi ih na drugi disk, uz sve ventilatore unutar kućišta koji donose nove vibracije itd. Sve ove vanjske vibracije (vibracije izvan samog diska) su u neprekidnom radu 365 dana (ili više), vrlo problematične odnosno stvaraju probleme u radu diska.

Sada postaje jasnije, zbog čega postoje diskovi za profesionalnu upotrebu.

Kod **SSD** diskova (engl. Solid State Disk) također postoji ista podjela prema *standardnim* i *profesionalnim* serijama diskova. Jedino su razlozi malo drugačiji. Prvi razlog je tip *Flash* memorije te njene pouzdanosti i broja ciklusa zapisivanja te njenih performansi. Drugi razlog je priručna brza RAM memorija (engl. *Cache*) na samom disku, uz koju se na *profesionalnim* SSD diskovima nalaze kondenzatori (učinak je poput *BBU* na RAID kontrolerima) za održavanje stanja memorije, do trenutka kada se svi podaci ne zapišu u *Flash* memoriju diska.

RAID kontroleri i RAID polja diskova

Ako smo se odlučili za RAID kontroler, svakako želimo imati hardverski RAID kontroler provjerenoga proizvođača. Dodatno, važna je i verzija *Firmware*a za RAID kontroler u kombinaciji s *driver*om za operativni sustav na kojemu ga koristite.
U praksi su se čak i kombinacije određenih verzija Firmware-a i *driver*a kao i kombinacije Firmware RAID kontrolera i Firmware-a drugih komponenti poslužitelja (pr. matične ploče), pokazale katastrofalnim odabirom.
Potrebno je dati si malo truda i proučiti što se kupuje, kao i komentare korisnika, za što približniju konfiguraciju vašoj : OS, *driver*i (RAID. LAN. MB, …), *Firmare*i i pripadajući Softver.

Cjenovni opseg poštenih RAID kontrolera (ovisno o broju diskova koje možete spojiti na njega), kreće se od minimalno tisuću KN na više. Sve ispod toga, kao i odabir Tzv. *Integriranih RAID* kontrolera na matičnim pločama (osim ako su u pitanju prave Serverske matične ploče cjenovnog ranga: 5.000+ KN) nemojte niti pomišljati koristiti.

Za više detalja pogledajte knjigu *Uvod u Linux i Linux napredno* - poglavlje "hardverski RAID"

Za više detalja pogledajte knjigu Uvod u Linux i Linux napredno - poglavlje ***hardverski RAID***

RAID polja diskova

U slučajevima kada želimo postići veću sigurnost, skalabilnost, brzinu pisanja ili čitanja na tvrdi disk ili SSD disk, potrebno je koristiti tehnologije u kojima se koristi više fizičkih diskova u istovremenom radu. Povezivanja više fizičkih diskova u neki logički disk odnosno polje diskova moguće je pomoću tzv. **RAID** tehnologija ili tzv. ***Volume Managera***.

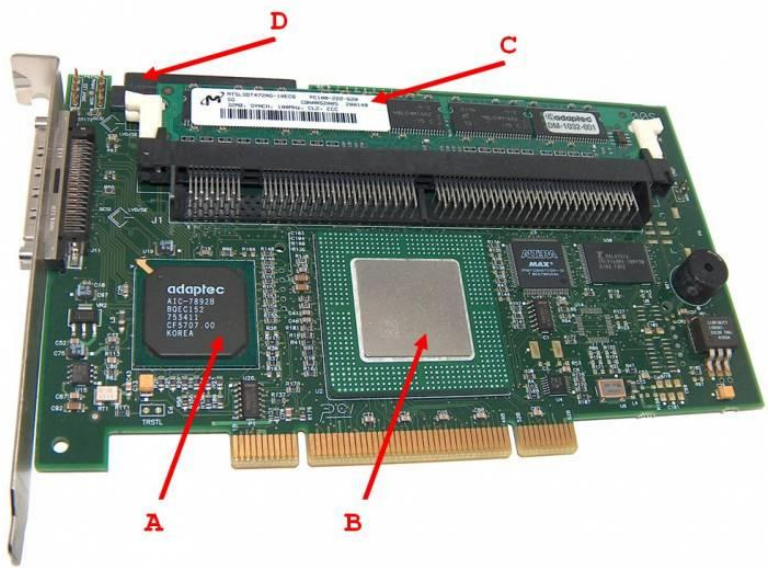
RAID je skraćenica od redundantnog polja nezavisnih diskova (engl. ***Redundand Array of Independent Disks***).

O čemu se radi ?
Podaci se zapravo zapisuju (i čitaju) na više diskova paralelno (istovremeno), na način koji je definiran u tzv. *RAID* polju, a koje smo odabrali prilikom inicijalizacije samog RAID polja. Postoji nekoliko standardnih RAID polja od kojih svako ima neke prednosti i mane.

Osim toga postoje dvije mogućnosti kreiranja RAID polja :

- ***Softverski*** - na razini operativnog sustava ili upravljačkog programa (*drivera*) ili
- ***Hardverski*** - za što je potreban RAID kontroler i pripadajući upravljački program za vaš operativni sustav.

Softversko RAID polje se kreira iz operativnog sustava a hardversko iz RAID kontrolera tj. posebne kartice poput ove na slici dolje



Na slici je RAID kontroler (Adaptec 2100S) nekadašnje tvrtke **Adaptec**, sada u vlasništvu tvrtke **Microsemi** .

Slika prikazuje hardverski RAID kontroler za spajanje SCSI tvrdih diskova pomoću 68 *pin*skog konektora/kabela, preko kojeg je moguće spojiti do 15 *SCSI* tvrdih diskova. Ovaj RAID kontroler se na matičnu ploču spaja preko PCI sabirnice.

Na njemu je vidljivo sljedeće:

- (**A**) SCSI disk kontoler (AIC-7892B)
- (**B**) CPU i RAID ASIC a koji (RAID ASIC) je zadužen za kreiranje RAID polja i druge operacije potrebne za RAID
- (**C**) RAM Memorija za cache podataka (u gornjem dijelu kartice) (32MB PC-100 ECC - proširivo)
- (**D**) 68pin-ski SCSI konektor za spajanje SCSI tvrdih diskova

Danas najčešće u upotrebi, imamo nekoliko osnovnih RAID polja, od kojih svako ima određene prednosti ali i neke mane.

Novije generacije *RAID* kontrolera, namijenjene su za spajanje **SATA** ili **SAS** diskova. Pogledajte sliku RAID kontrolera tvrtke **LSI** (sada u vlasništvu tvrtke **Broadcom**), koji se spaja na PCIe x8 sabirnicu te omogućava spajanje do 8 SAS ili SATA diskova (Oznaka modela je *Broadcom LSI MegaRAID SAS 9361-8i*).

Izvor slike: Tvrtka Broadcom Limited

Standardna RAID polja

Važno za znati, je to, što hardverski ili softverski RAID kontroleri kod inicijalizacije RAID polja, logičku površinu svih diskova u polju, dijele na male blokove podataka (ovisno o RAID kontroleru i/ili softveru).

Ove male logičke cjeline postaju najmanje jedinice za zapis podataka, praktično kao što su *clusteri* tj. blokovi na datotečnom sustavu. Veličina ovih blokova podataka u RAID terminologiji se zove *Stripe size*.

Svi diskovi u nekom RAID polju moraju imati istu podjelu na najmanje blokove podataka, koji se mogu čitati ili pisati tj. nad njima provoditi određene logičke operacije. Najčešća veličina *Stripe size* je 64kB ali ju je moguće i mijenjati prije nego se krene u proces inicijalizacije RAID polja.

Ne preporuča se mijenjati *Stripe size* ukoliko stvarno ne znate zašto to radite

Standardna RAID polja su:

- **RAID 0**
- **RAID 1**
- **RAID 5**

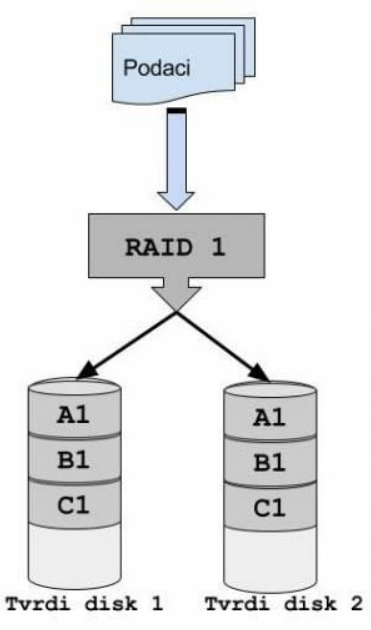
Ugniježđena RAID polja su:

- **RAID 01**
- **RAID 10**
- **RAID 50**
- **RAID 60**
- **RAID 100**

RAID 0

RAID 0 (Engl. *Stripe*) polje sastoji se od minimalno dva (2) diska. Ukupan kapacitet ovog polja, jednak je zbroju kapaciteta svih diskova u polju. U ovo polje moguće je dodavati diskove različitih veličina ali polje će se uglavnom kreirati tako, da će na svim diskovima biti iskorišten samo kapacitet od najmanjeg diska.

Ako imamo diskove od 120 GB , 320 GB i 500 GB, ukupni kapacitet će biti 120 GB x 3 = 360 GB ([na većini RAID kontrolera]).



Zapisivanje podataka u **RAID 0** polje radi se tako, da se podaci koji se trebaju zapisati dijele na blokove (pr. 64kB) i pri tome se razlomljeni blokovi podataka pravilno zapisuju (raspoređuju) na sve diskove u polju.

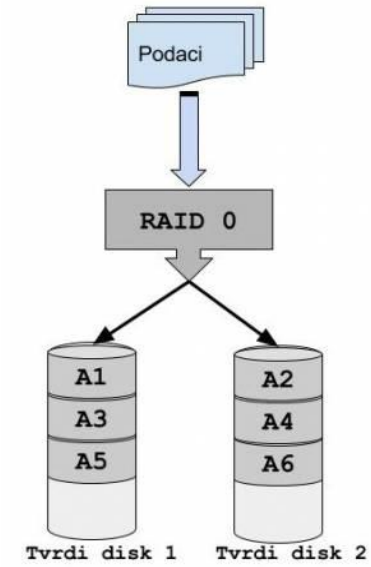
Zamislimo kako imamo dva diska kao na slici. Podaci se tada raspoređuju i zapisuju, pola na jedan a pola na drugi disk i to redom kako dolaze.

Ako su recimo blokovi (*Stripe size*) od 64kB na svakom disku kao najmanje jedinice za zapisivanje i imamo 384kB podataka za zapisati, došaša se slijedeće:

- prvih 64kB će biti zapisano na prvi disk (*A1*)
- drugih 64kB na drugi disk (*A2*)
- trećih 64kB na prvi disk (*A3*)
- četvrtih 64kB na drugi disk (*A4*)
- petih 64kB na prvi disk (*A5*)
- šestih 64kB na drugi disk (*A6*)

RAID 1

RAID 1 (Engl. *Mirror*) odnosno zrcalno polje, sastoji se od minimalno 2 diska. U ovom polju moguće je imati i više od 2 diska uz isto ograničenje za kapacitet kao za RAID 0 (kapacitet najmanjeg diska se koristi na svim diskovima). Ovo polje radi identičnu (zrcalnu) kopiju svih podataka s prvog diska na drugi (ili na više njih) i to također kako podaci dolaze, razbijanjem u blokove (engl. Stripe).



Ako imamo 2 diska u RAID 1 polju i zapisujemo pr. 192kB podataka događa se sljedeće:

1. prvih 64kB se zapisuje na prvi i drugi disk (A1)
2. u drugom koraku se drugih 64kB zapisuje na prvi i drugi disk (B1)
3. u trećem koraku se trećih 64kB zapisuje na prvi i drugi disk (C1)

Ovime imamo zapisano svih 192kB na oba diska (pogledajte sliku lijevo).
Ovo polje daje nam najveću sigurnost jer su svi podaci zapisani na prvi disk zapisani i na drugi disk (ili koliko ih već ima u polju).
Brzina čitanja je veća jer se podaci mogu čitati u parovima (polovina podatka s prvog a pola s drugog diska).

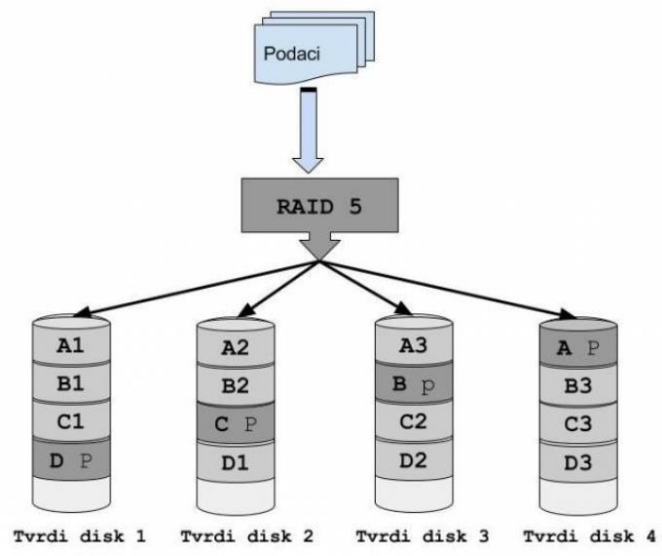
Na brzinu čitanja najviše utječe implementirana funkcionalnost čitanja s više diskova paralelno što ovisi o RAID kontroleru tj. softveru koji to odrađuje (lošije implementacije daju i lošije rezultate).
Brzine kod zapisivanje su jednake brzini jednog diska jer se podaci moraju istovremeno zapisivati na sve diskove u polju.

RAID 5

Za ovo RAID polje potrebna su nam minimalno 3 tvrda diska. Ovo je prvo specifično polje koje koristi računanje pariteta niza podataka, koje je procesorski zahtjevnije od RAID 0 i RAID 1 polja. Ovdje do izražaja dolaze hardverski RAID kontroleri koji imaju poseban procesor (CPU) koji izračunava paritet za svaki niz podataka.

Paritet je podatak, na osnovu kojeg, se unutar svakog niza podataka, u slučaju gubitka jednog od blokova podataka tog niza, može povratiti izgubljeni blok podataka.

Zamislimo **RAID 5** polje sa 4 tvrda diska kao na slici.



Uzmimo standardni *Stripe size* od 64kB. U ovom polju, sa 4 tvrda diska svaki niz podataka se lomi na 3 dijela (ukupan broj diskova -1). Zamislimo da moramo zapisati 768kB podataka. U prvom nisu se zapisuje:

- prvih 64kB na prvi disk (**A1**)
- drugih 64kB na drugi disk (**A2**)
- trećih 64kB na treći disk (**A3**)
- Potom se radi logička XOR operacija prvih, drugih i trećih 64kB podataka i dobija se rezultat (engl. Parity), koji je veličine 64 kB i zapisuje se na četvrti disk (**A P**).

U sljedećem nizu podataka razlomljeni podaci se zapisuju na prvi (**B1**), drugi (**B2**) i četvrti disk (**B3**) a paritet se izračunava i zapisuje na treći disk (**B P**).

Kod svakog sljedećeg niza podataka, paritet se izračunava i zapisuje na neki drugi disk, tako da ako nam se pokvari bilo koji disk u polju, od preostalih diskova se pomoću paritetnih podataka mogu izračunati (restaurirati) podaci koji nedostaju.

Kako radi računanje pariteta (XOR funkcija) ?

Prvo se podsjetimo XOR logičke operacije

Ulaz A	Ulaz B	XOR rezultat
0	0	0
0	1	1
1	0	1
1	1	0

Dakle ako su oba ulaza jednaka, rezultat je nula (0) a ako su ulazi različiti, rezultat je jedan (1).
Vratimo se na naša 4 diska i *Stripe*ove od 64kB. Pošto nam je 64kB preveliki broj, za razumijevanje logike rada, zamisliti ćemo da je *Stripe size* samo 4 bita.
Zamislimo da moramo zapisati sljedeći podatak : `101111100110`.
Pošto se on mora razložiti na 3 dijela po 4 bita (4. bita nam je *Stripe size*) to će izgledat ovako : `1011 1110 0110`.
Sada svaki od ovih razlomljenih dijelova ide na zapisivanje, svaki na svoj disk :

1. Podaci za disk 1 : `1011`
2. Podaci za disk 2 : `1110`
3. Podaci za disk 3 : `0110`

Radi se XOR prvi disk: `1011` i drugi disk: `1110`, i dobivamo rezultat `0101`
Sada se radi XOR prvog rezultata: (`0101`) sa trećim diskom i dobivamo paritetni podatak: `0011`

Dakle na četvrti disk se zapisuje paritet : **0011**
Sada imamo sljedeće zapise u prvom nizu zapisivanja:

Disk 1	Disk 2	Disk 3	Disk 4
1011	1110	0110	0011

…i tako redom, drugi niz, pa paritet na disku 3, pa sljedeći niz s paritetom na disku 2, pa sljedeći niz s paritetom na disku 1, … … …

Kvar pojedinog diska

Zamislimo da nam se pokvario **Disk 2** te imamo slijedeće stanje:

Disk 1	Disk 2	Disk 3	Disk 4
1011	_ _ _ _	0110	0011

Sada se ponovno radi XOR operacija, redom (Disk1, Disk 3, Disk 4):

- Disk 1: **1011** XOR Disk 3: **0110** - rezultat je **1101**
- Rezultat prvog XOR: **1101** XOR Disk 4: **0011** - rezultat je **1110**
- Dobili smo rezultat odnosno sadržaj bloka tj. *Stripe*a koji je neispravan : **1110** , što je točno.

Što se još događa kada imamo neispravan disk ?
Ako nemamo *Hot Spare* disk ili nismo zamijenili neispravan disk, RAID kontroler će raditi XOR u radu (u letu), za svaki niz podataka, tako da će sve raditi i dalje (uz djelomično usporenje) ali ako nam se istovremeno pokvari još jedan disk, izgubiti ćemo **SVE** podatke.

Za RAID polja: 1, 5, 6, 10, 50 i 60
Ako smo ipak imali *Hot Spare* disk, RAID kontroler će započeti s procesom restauriranja podataka (engl. *Rebuild*) na *Hot Spare* disk. Ako nismo imali *Hot Spare* disk, kada izvadimo neispravni disk i ubacimo ispravni, proces restauriranja će se pokrenuti. Ovaj proces kreće uglavnom automatski, u radu. Pošto sada RAID kontroler mora preračunati (ili prerasporediti - ovisno o RAID polju) podatke za svaki niz podataka, na cijeloj površini svih tvrdih diskova. Ova procedura može potrajati satima ili čak danima - u slučaju da imamo velike količine podataka, a i ovisno o RAID polju.

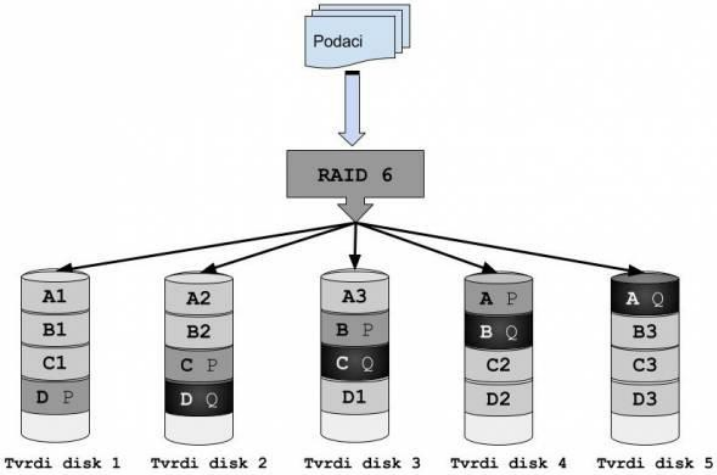
Karakteristike RAID 5 polja

Brzina čitanja je prilično velika jer se podaci mogu čitati s više diskova istovremeno. Brzina pisanja je manja od RAID 0 i RAID 1 polja jer se podaci zapisuju na više diskova paralelno ali se za svaki niz podataka mora izračunavati i paritet.

Ukupni kapacitet cijelog polja je jednak zbroju svih diskova minus jedan (-1) disk. RAID 5 polje dozvoljava ispad samo jednog tvrdog diska iz polja i to bilo kojeg diska u polju.

RAID 6

Za ovo RAID polje potrebna su nam minimalno 4 tvrda diska. Ovo polje je slično RAID 5 polju s time da se ovdje računaju dva (2) pariteta koja se razmještaju na različite tvrde diskove u polju. Zamislimo osnovno RAID 6 polje poput ovoga na slici



Paritet se razmješta po istom principu kao kod RAID 5 polja, svaki puta se za svaki novi niz podataka pomiče na druge diskove. S razlikom što ovdje imamo dva pariteta od kojih se svaki zapisuje na svoj disk. Za prvi niz podataka, prema tome imamo (ako je *Stripe size* 64kB):

1. prvih 64kB se zapisuje na prvi disk (**A1**)
2. drugih 64kB se zapisuje na drugi disk (**A2**)
3. trećih 64kB se zapisuje na treći disk (**A3**)
4. prvi paritet od 64kB se zapisuje na četvrti disk (**A P**)
5. drugi paritet od 64kB se zapisuje na peti disk (**A Q**)

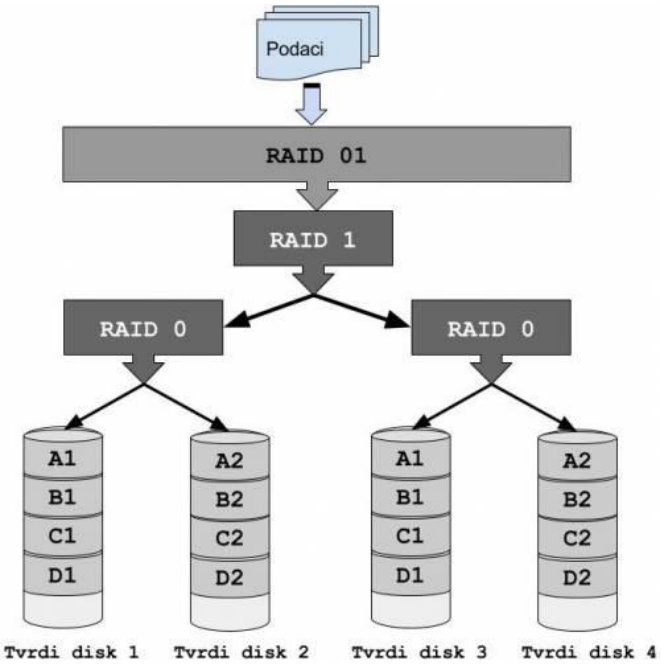
Dalje se sve mijenja slično kao kod RAID 5, s ciljem da se za svaki slijedeći niz podataka, pariteti zapisuju na druge diskove.

Ovo polje diskova stoga omogućava ispad do dva bilo koja tvrda diska u polju.

Ukupni kapacitet polja jednak je zbroju kapaciteta svih diskova minus kapacitet dva diska (-2). Brzina čitanja je slična ili nešto sporija kao kod RAID 5 polja (ovisno o implementaciji).

Brzina pisanja također ovisi o implementaciji (hardver [i to koji ?] / softver) a također je slična ili lošija od RAID 5 zbog dvostrukog izračunavanja pariteta.

Ugniježđena RAID polja (RAID 0+1)



Ovo RAID polje često se naziva i **RAID 01**. Ovo polje (engl. Mirror of Stripes) nastaje kreiranjem dva ili više **RAID 0** polja na dva ili više diskova koja se stavljaju unutar većeg RAID 1 polja. Konačno **RAID 1** polje se prijavljuje kao logički disk *BIOS*u matične ploče računala.

Kapacitet je jednak polovici kapaciteta svih diskova. Ovo polje, otporno je na ispad cijelog jednog **RAID 0** polja (pošto **RAID 0** ispada uslijed kvara samo jednog diska), a **RAID1** (engl. Mirror) zrcaljenje štiti s identičnom kopijom, koja se nalazi na drugom **RAID 0** polju. U slučaju ispada dva diska iz različitih **RAID 0** polja, cijelo **RAID 01** polje se ruši i gube se **SVI** podaci. Proširivanje ovog polja (kapacitetom) je nepraktično (i skoro neizvedivo u praksi) jer zahtjeva manipulacije na svim **RAID 0** poljima.

Brzina čitanja je prilično velika pošto se podaci mogu čitati sa svih diskova istovremeno (dio podataka s a svakog diska). Brzina pisanja je također prilično velika pošto se podaci mogu čitati sa svih diskova istovremeno (dio podataka sa svakog diska).

Za rad ovog polja potrebno je minimalno 4 diska, kao što je vidljivo na slici

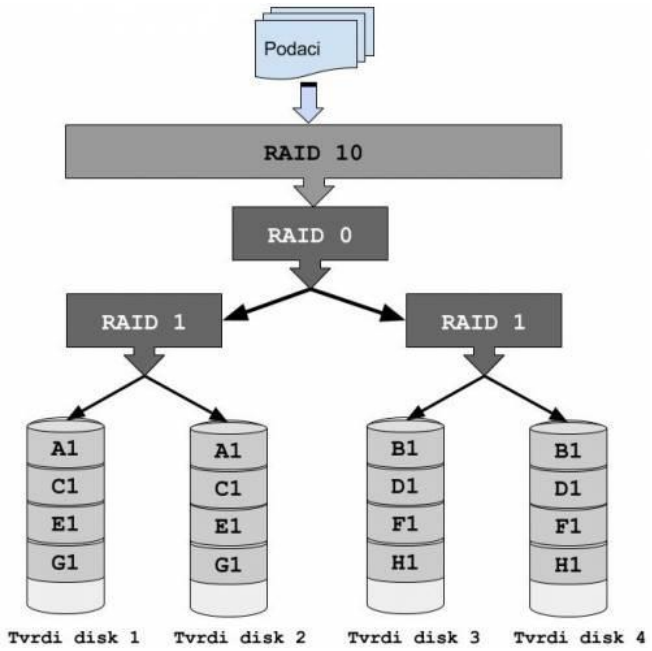
Ugniježđena RAID polja (RAID 1+0)

Ovo RAID polje često se naziva i **RAID 10**. Za rad ovog polja potrebno je minimalno 4 diska Ovo polje je slično **RAID 01** ali prema logici rada, potpuno obrnuto. Po dva diska se dodaju u **RAID 1** polje (engl. Mirror) odnosno zrcalno se kopiraju jedan na drugi. Druga dva diska također, u svoje **RAID 1** polje, a isto tako i treća, četvrta , peta ili x-ta dva diska. Potom se kreira **RAID 0** (engl. Stripe) koji povezuje sve zrcaljene parove diskova u jedan veliki logički disk.

Ispadom (kvarom) od po jednog diska iz svakog RAID 1 polja, nemamo problem. Jedino ispadom oba diska unutar pojedinog RAID1 polja gubimo sve podatke. Dakle ovo polje, otporno je na ispad bilo kojeg diska unutar bilo kojeg RAID1 (engl. Mirror) zrcalnog para diskova, na bilo kojem RAID 1 paru diskova.

Kapacitet je jednak polovici kapaciteta svih diskova. Brzina čitanja je prilično velika pošto se podaci mogu čitati sa svih diskova istovremeno (dio podataka sa svakog diska). Brzina pisanja je također prilično velika pošto se podaci mogu čitati sa svih diskova istovremeno (isto dio podataka sa svakog diska). Dodatna prednost je i u tome što neki RAID kontroleri omogućavaju proširenje kapaciteta vršnog RAID 0 polja (koje je vidljivo kao logički disk) dodavanjem novog para RAID 1 (engl. Mirror) diskova.

Primjer rada RAID 10 polja je vidljiv na slici



Kako se zapisuju podaci ?. Ako imamo 512kB podataka (uz standardni *stripe size* od 64 kB) koji se trebaju zapisati, događa se slijedeće:

1. u prvom koraku se prva 64kB kopiraju i paralelno zapisuju na oba diska u prvom **RAID 1** polju (**A1** i **A1**)
2. u drugom koraku se druga 64kB kopiraju i paralelno zapisuju na oba diska u drugom **RAID 1** polju (**B1** i **B1**) jer je ovo drugo **RAID 1** polje zapravo *Stripe* odnosno **RAID 0** polje, prvom **RAID 1** polju.
3. u trećem koraku se treća 64kB kopiraju i paralelno zapisuju na oba diska u prvom **RAID 1** polju (**C1** i **C1**)
4. u četvrtom koraku se četvrta 64kB kopiraju i paralelno zapisuju na oba diska u drugom **RAID 1** polju (**D1** i **D1**) jer je ovo drugo **RAID 1** polje zapravo *Stripe* odnosno **RAID 0** polje, prvom **RAID 1** polju.
5. … i tako redom dok se ne zapišu svi podaci

Osim navedenih ugniježđenih RAID polja, moguća su i polja **RAID 100**, **RAID 50** i **RAID 60**

Integrirana RAID rješenja

U ovim *Integriranim* kombinacijama, dobivate upravljački program (driver) i pripadajući softver, koji :

- je u pravilu loše dokumentiran ili
- se jako rijetko održava ili
- je pun grešaka
- a u slučaju katastrofe nema definirane korake (što i kako) ili su oni nejasni (a često i ne rade)

Kod ovih rješenja, zapravo ne postoji pravi hardverski RAID kontroler, već se većina ili gotovo sve funkcionalnosti RAID kontrolera odrađuju unutar upravljačkog programa (drivera) koji se *pretvara* da je RAID kontroler. On nadalje operativnom sustavu prijavljuje RAID polje diskova o kojem se brine, kao jedan fizički disk - slično kako bi to uradio i pravi RAID kontroler.

I na kraju sve je zapravo prepušteno navedenom softveru (driveru) i operacijskom sustavu, pa što bude. Nakon nekog vremena obično bude : **`Nešto ne radi. A kako da vratimo svoje podatke`** … e lijepo sam vam rekao.

Na kraju priče s krivim odabirom tehnologije ili uređaja uvijek završite sa slanjem diskova u neku od tvrtki specijaliziranih za povrat podataka. Ova zabava će vas obično koštati znatno više nego da ste odmah kupili možda i najskuplje komponente koje postoje na našem tržištu.

Povrat odnosno spašavanje podataka se često naplaćuje po GB pa cifre vrlo brzo narastu na desetke tisuća KN.

Opcija dva : Logical Volume Manager

Iz mog iskustva - zaboravimo na *Integrirane RAID* kontrolere. Druga prihvatljivija opcija je upotreba Tzv. *Logical Volume Manager*a unutar operacijskog sustava (govorimo o Linuxu, mada slična rješenja postoje i na drugim operacijskim sustavima). Naime čak i Windows OS nudi razna softverska RAID polja, koja su pouzdanija od *integriranih RAID* rješenja.

U Linuxu se radi o *Logical Volume Manager*u verzije 2 (LVM2). Za više detalja o LVM2 pogledajte Knjigu *Uvod u Linux i Linu napredno*, poglavlje LVM2.

U Windowsima, ovisno o verziji (7, 8, 10 ili Server 2008/R2, 2012/R2, 2016) imamo ponuđeno nekoliko RAID polja:

- ***Spanned Volume*** - slično RAID 0 polju ali zamišljeno za proširenje kapaciteta, dodavanjem novih diskova, bez sigurnosti uslijed kvara bilo kojeg diska (poput RAID 0)
- ***Striped Volume*** - klasično RAID 0 polje
- ***Mirrored Volume*** - klasično RAID 1 polje
- ***RAID-5*** - klasično RAID 5 polje

U navedenim slučajevima, radi se o softverskom *RAID*u odnosno njegovoj funkcionalnosti. Ipak ovo je puno sigurnije i stabilnije riješenje od onoga koje dobivamo s upotrebom *Integriranih RAID kontrolera* na matičnim pločama za stolna računala (tj. Ne serverskim matičnim pločama). Ovo je dokazano rješenje koje koristi velika zajednica ljudi, a koje se razvija s otvorenim kodom (govorimo o **LVM2**) pa se svaka novootkrivena greška vrlo brzo popravlja. Osim toga vrlo je dobro dokumentirano (vrijedi i za Windows riješenje i **LVM2** pogotovo) te postoje dobro definirane procedure u slučaju neke havarije odnosno što i kako napraviti u kojem slučaju.

Zbog čega je ipak bolji hardverski RAID kontroler i pripadajuće RAID polje

Zbog toga što će vam na budućem NAS ili SAN sustavu biti pohranjeni važni podaci. Stoga ne želite da se sve snima na samo jedan tvrdi disk, već na više njih i to u RAID polju koje vam osigurava najbolji odnos:

- sigurnosti podataka (koliko kopija podataka želite - na dva, tri ili više tvrdih diskova istovremeno)
- brzine rada
- brze zamjene neispravnog diska i povratka u normalan rad
- lakoće proširenja kapaciteta RAID polja

Dodatno želite **RAID** kontroler jer ne želite prepustiti nekom traljavom programčiću ili sustavu (mislim na *Integrirana RAID rješenja*), da vam odrađuje ovako važne zadatke pohranjivanja vama osjetljivih i za život tvrtke važnih podataka. A s druge strane želite iskoristiti

hardversku snagu pravog RAID kontrolera koji ne opterećuje sustav (CPU/RAM) pošto ima svoj specijalizirani CPU i pripadajuću RAM memoriju.

Ozbiljan zadatak oko *RAID*a prepustimo tvrtki koja RAID **kontrolere proizvodi profesionalno, kao i njihov pripadajući firmware/softver.**

Neka najbolji proizvođač specijaliziranog hardvera i softvera zaradi na onome što radi najbolje.

Druga opcija je uvijek, upotreba:

- **LVM2** sustava - na Linuxu ili
- Windows softverskog ekvivalenta RAID poljima ili
- nekog drugog profesionalnog rješenja, kao što je primjerice **ZFS** (o njemu malo kasnije)

Odabir RAID polja

Kao što je i odabir dobrog RAID kontrolera važan, važno je i pravilno odabrati RAID polje, ovisno o vašem budžetu i potrebama.

Podsjetimo se RAID polja o kojima smo govorili

Skraćena verzija (navesti ćemo par često korištenih RAID polja) :

RAID polje	**Brzina**	**Min. broj diskova**	**Koliko diskova se može istovremeno pokvariti**	**Vrijeme od zamjene pokvarenog diska do normalnog rada**	**Jednostavnost proširenja RAID polja**
1	Jednako ili brže od jednog diska	2	1	Brzo	Vrlo teško
5	Brže od RAID 1	3	1	Sporo	Teško
6	Sporije od RAID 5	4	2	Sporo	Teško
10	Najbrže	4	2	Brzo	Jednostavnije

I na kraju zbog čega priča o diskovima, RAID kontrolerima i RAID poljima ?

Kada smo kreirali neko RAID polje (na RAID kontroleru po mogućnosti) slijedeće je na redu upotreba tog RAID polja, preko nekog od mrežnih protokola za **NAS** (NFS, SNB/CIFS, AFP ili dr.) ili **SAN** sustav, a preko kojih dijelite podatke preko mreže (vrlo jednostavno).

Dakle kreiranje RAID polja je prva od faza kod stvaranja nekog **NAS** ili **SAN** sustava.

Priča o particijama i datotečnom sustavu

U ovom poglavlju, na RAID polje (kakvo god bilo), gledati ćemo sa više razine, odnosno sa strane operacijskog sustava. Pretpostaviti ćemo, da ste upoznati s geometrijom diskova, pa ćemo samo reći da se podaci u konačnici, na disk, zapisuju u **cluster**e, koji su najmanje jedinice zapisa na svakom disku.

Ako niste upoznati s geometrijom diskova, pogledajte knjigu : **Uvodu Linux i Linux napredno** - poglavlje **Geometrija diskova**

Dalje u tekstu, govoriti ćemo o Tzv. *PC stilu* odnosno vrsti particija, s kojima ćete se najčešće i susretati.

I dalje ćemo govoriti o **PC stilu** particioniranja i to o **MBR** vrsti, zbog lakšeg razumijevanja, a koja predstavlja stariji standard (potječe iz 1983.g). MBR radi na sličnom principu kao i noviji standard, koji se zove **GPT** (*GUID Partition Table*).

MBR sustav:

- Unutar MBR zapisa, nalaze se popisane particije kao i početni i završni sektor, svake particije. Ovaj prostor, koji služi za adresiranje svake particije je 32-bitni broj. Dakle maksimalna veličina particija prema *MBR* shemi je, prema tome 2^{32} za 512*byte* (veličina sektora) = 2 TB. Detaljnije radi se o adresiranju koje se zove **LBA** (Logical Block Addressing). Uglavnom, to znači da ne možemo imati particiju veću od **2TB**. Netko se može zapitati: što ako imamo noviji tvrdi disk, kojemu veličina sektora nije 512 *byte*a nego 4096 (što je slučaj sa svim najnovijim (velikim) diskovima). Odgovor je MBR podržava samo 512 *byte* sektore.
- MBR podržava maksimalno 4 primarne particije. Veći broj particije je moguće imati ako na mjestu primarne particije napravimo Tzv. *extended* odnosno proširenu particiju, unutar koje je moguće kreirati više logičkih particija

GPT sustav:

- Maksimalni broj particija je toliki da praktično nećete imati ograničenja na njihov broj (do 128 primarnih particija).
- Maksimalna veličina pojedine particije je sada 64 *bitni* broj. Dakle za 512 *byte* sektore je to 2^{64} *byte*, za adresiranje tj. 9.4 **ZB** (*Zetta Byte*a). A za 4096 *byte* sektore (tj. 4KB) je to 2^{72} *byte*a za adresiranje tj. 19 **YB** (***Yotta Byte***). To znači kako limitirajući faktor postaje datotečni sustav i njegove mogućnosti.
- Veličina sektora može biti i 512 *byte*a i 4096 *byte*a**.**

S **MBR** particijama možemo raditi sa svim programima za particioniranje, dok sa **GPT**, samo s novijima, koji podržavaju **GPT**particije.

Bez obzira, koristimo li **MBR** ili **GPT** shemu, svaki novi disk je potrebno pripremiti za korištenje, što podrazumijeva:

1. **Particioniranje** (kreiranje minimalno jedne particije na disku)
2. **Formatiranje** (instaliranje datotečnog sustava, na određenu particiju)

Particija predstavlja logičku cjelinu tvrdog diska, koju operativni sustav može koristiti. Particija je i najmanja logička cjelina, na koju se može instalirati datotečni sustav.

Naime datotečni sustav (NTFS, ext2/3/4, XFS, ZFS i sl.) se može instalirati samo na particiju na disku, pa prema tome, sve dok disk nije podijeljen na particije (minimalno jednu), na njega se ne može instalirati datotečni sustav. Definicija svih particija na disku se nalazi u particijskoj tablici (Engl. partition table), koja se nalazi u **MBR** zapisu.

Detaljnije, MBR (*Master Boot Record*) sadrži informacije o particijama i njihovim pozicijama (na kojoj lokaciji počinje i završava koja particija). Pozicija na prvom sektoru particije naziva se **Boot Record** (BR), ili **Volume Boot Record** (VBR).

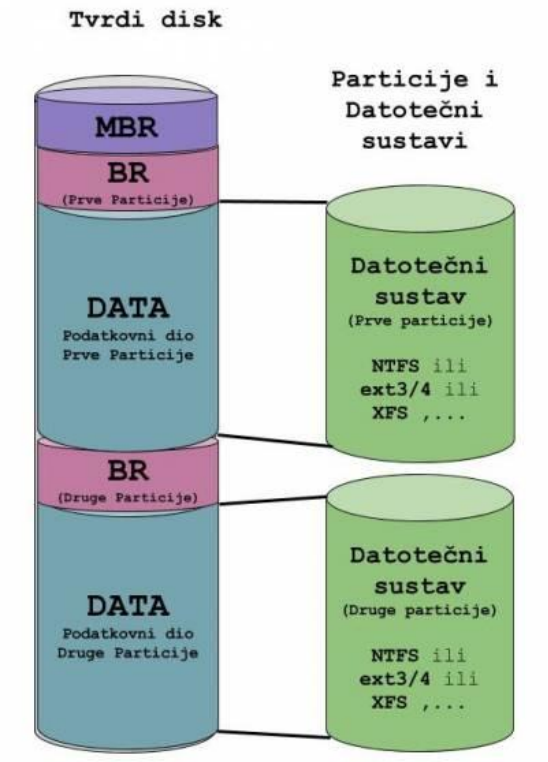
Svaki disk mora imati minimalno jednu particiju.
Formatiranje se odnosi na instalaciju datotečnog sustava na odabranu particiju. Dakle nakon što smo kreirali particiju na disku, potrebno je instalirati datotečni sustav, ovisno o operativnom sustavu, na kojem to radimo. Neki od datotečnih sustava su:

- **NTFS** - za Windows operativne sustave (prepoznaje ih i Linux)
- **ext2, ext3, ext4, XFS ,ZFS** - za Linux operativne sustave

Kada smo instalirali datotečni sustav na odabranu particiju, disk je spreman za upotrebu.
Bez obzira da li imamo jedan disk ili se radi o RAID polju diskova (koje je ionako vidljivo kao jedan disk), procedura je ista:

1. **Particioniranje**
2. **Formatiranje**

Pogled na *MBR* shemu, ako gledamo površinu jednog diska, s dvije particije.



Kako je vidljivo na slici, svaki disk na prvom sektoru (na početku diska) ima **MBR** (*Master Boot Record*) zapis, u kojemu se nalazi popis svih particija i njihove lokacije.

Svaka particija, na svom početku ima **BR** (*Boot Record*), koji se nekada zove i **VBR** (*Volume Boot Record*) zapis,a u kojemu se obično nalazi Tzv. *second stage loader*, za učitavanje operativnog sustava (ako ga imamo na toj particiji).

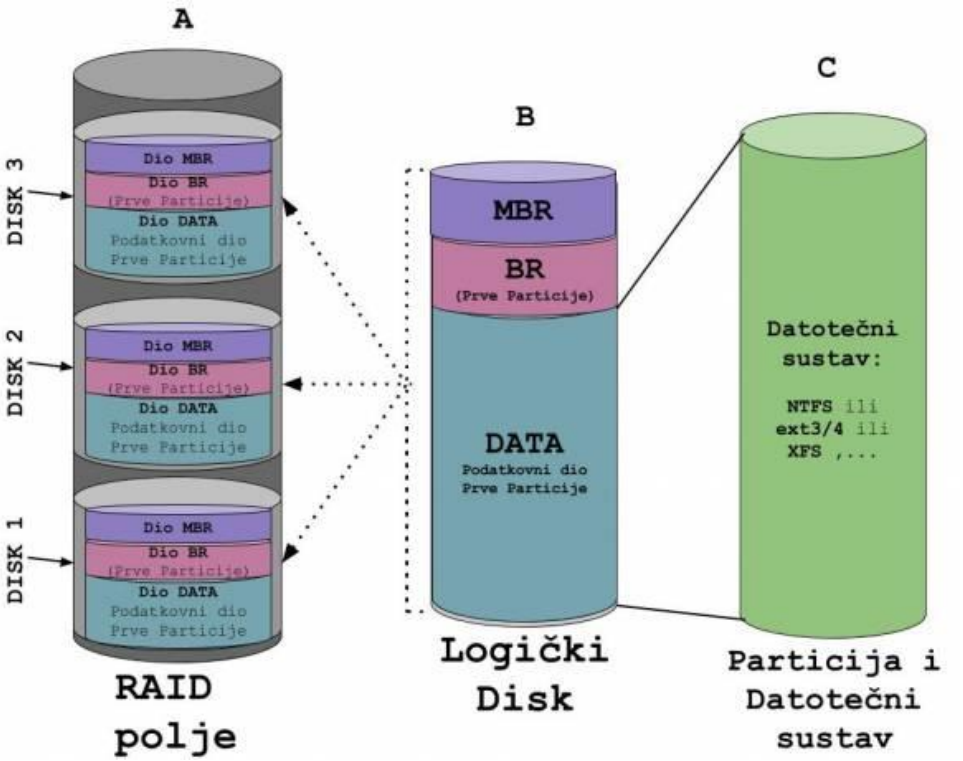
Ovakva struktura se često koristi kada imamo potrebu za instalacijom dva (ili više) operativnih sustava, svakog, na svojoj particiji.

U tom slučaju particija s koje se pokreće operativni sustav, mora biti označena kao *boot*abilna. Tada se prva faza *boot loade*ra nalazi u **MBR** zapisu a druga faza tj. *second stage boot loader* za operacijski sustav se nalazi u ***BR*** zapisu, na svakoj particiji.

Druga moguća upotreba je jednostavno kod primjena u kojima želimo razgraničiti i razdijeliti disk na više particija, na kojima ćemo imati različite stvari : pr. na prvoj particiji že biti operativni sustav a na drugoj podaci. Ili će na obije particije biti podaci ali za različite namjene.

Nakon prvog sektora na particiji se nalazi *DATA* dio, koji možemo formatirati s željenim datotečnim sustavom (NZFS, ext2/3/4, XFS, ZFS, …).

Pogledajmo sada stanje u kojemu koristimo RAID kontroler s jednom particijom (*MBR* shema):



Sa strane RAID kontrolera (pogled **A**), vidljivi su svi diskovi u RAID polju. U RAID polju se podaci raspoređuju po svim diskovima, ovisno o samom RAID polju. Tako je i sam *MBR* raspoređen (kopiran) na više diskova u RAID polju. Isto se događa i s *BR* i svim podacima (DATA dio).

Nadalje RAID kontroler prijavljuje *BIOS*u računala cijelo RAID polje kao jedan (običan) (logički) disk (pogled **B**). Ako sve pogledamo s točke tog logičkog diska (pogled **B**) on se ponaša kao običan disk, koji se sastoji od *MBR*a, *BR*a i dijela za podatke (DATA dio).

Pogledamo li ipak sve iz točke operativnog sustava, sa strane korisnika (pogled **C**), dostupna nam je particija diska (DATA dio), na koju možemo instalirati datotečni sustav, po želji : **NTFS, ext2/3/4, XFS, ZFS** ili koji već trebamo.

Što je uopće NAS sustav ?

NAS (Engl. *Network Attached Storage*) odnosno *mrežno spojena spremišta podataka* osiguravaju nam prostor za spremanje podataka, preko mreže. Ovo su zapravo mrežni dijeljeni sustavi za spremanje podataka. Oni rade na razini datoteka (i naravno direktorija), koje pohranjujemo na njih i to preko mrežnih protokola za dijeljenje datoteka.

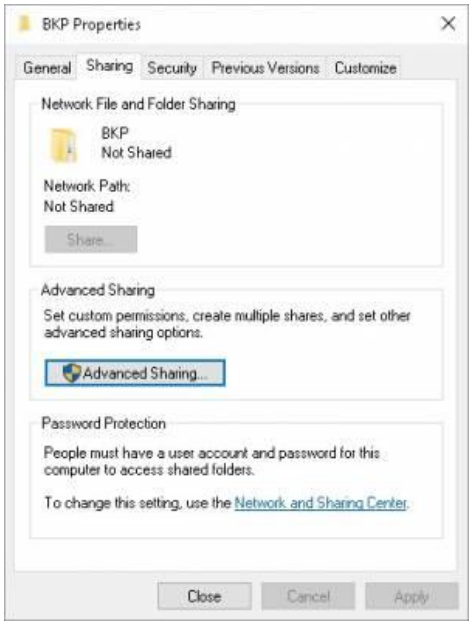
Možemo reći da je NAS sustav, sustav odnosno računalo ili poslužitelj, koji preko mreže daje pristup svom diskovnom prostoru, na razini datoteka i direktorija. Ovakav sustav je, prema tome, specijaliziran za dijeljenje datoteka s klijentima odnosno drugim računalima na mreži. Ovakav sustav može biti izveden hardverski ili softverski, a uglavnom je kombinacija jednog i drugog. Dodatno, ovakvi sustavi na sebi, obično imaju ugrađeno po nekoliko tvrdih diskova (ili **SSD** u bržoj varijanti) a koji se uglavnom nalaze u nekom **RAID** polju.

Kako smo naučili, to RAID polje je prvo potrebno *particionirati* te formatirati sa željenim datotečnim sustavom. Taj datotečni sustav se dalje koristi za pohranu podataka, preko nekog od protokola za mrežno dijeljenje datoteka, te tako postaje NAS sustav.

Svako dijeljenje datoteka preko mreže (Engl. Network Share), korištenjem nekog od mrežnih protokola koji postoje za tu namjenu, možemo nazvati upotrebom kao NAS sustava.

U konačnici, na svakom tom **NAS** sustavu (računalu/poslužitelju) se preko mreže, dolazi do dijeljenog pristupa datotekama pomoću mrežnih protokola za tu namjenu. Navesti ćemo one najčešće u upotrebi:

- **NFS** (Network Files System) - koristi se uglavnom na Linux/Unix operacijskim sustavima (ili ponekad u Windows okruženju). Open source varijanta podrazumjeva korištenje nekog od **nfs** daemona (servisa)
- **SMB/CIFS** (Server Message Block / Common Internet File System) - koristi se uglavnom na Windows ili Linux okruženjima. Koristi se osim za dijeljenje datoteka i za dijeljenje pisača, i drugih uređaja te dodatnih funkcionalnosti,preko mreže.
    - Open source riješenje se zove samba
    - **Windows Share** je integriran u sve Windows operativne sustave.
- **AFP** (Apple Filing Protocol) - koristi se za dijeljenje datoteka na Mac OS računalima.
- Istoj kategoriji pripadaju i **FTP** (File Transfer Protocol) i **TFTP** (Trivial File Transfer Protocol) protokoli, s time da su oni jednostavniji i nemaju naprednije mogućnosti kao gore navedeni.
- Često se koristi i **WebDAV** (Web Distributed Authoring and Versioning) koji je što se tiće funkcionalnosti negdje između FTP i gore navedenih protokola
- …



Najosnovniji primjer upotrebe sustava za koji bi mogli reći da je neka vrsta **NAS** sustava bi bilo klasično dijeljenje nekog direktorija preko mreže, iz **Windows** operativnog sustava (vjerujem da je to svima poznato).

Nešto poput dijeljenja (Engl. Sharing) ovog direktorija (D:\BKP) na slici .

U slučaju upotrebe na **Windows** operacijskom sustavu, sve se za jednostavno mrežno dijeljenje svodi na odabir željenog direktorija te njegovog dijeljenja preko SMB/CIFS mrežnog protokola.

Osnovne pretpostavke i planiranje za NAS ili SAN sustav

U slučajevima kada ste se odlučili sami “napraviti” NAS ili SAN sustav, važno je znati da tom procesu prethodi malo proučavanja i odabira.

1. Kod odabira NAS ili SAN sustava odnosno poslužitelja, prvo moramo biti svjesni zahtjeva za softverom (operatvnim sustavom). Operativni sustav za NAS ili SAN može biti :

- Specijalizirani *open source* OS poput :
  - ***OpenFiler***
  - ***OpenMediaVault***
  - ***FreeNAS***
  - ***Nas4Free***
  - ***NexentaStor*** **(*Nexenta* Community Edition)**
- ili neko od komercijalnih rješenja za NAS/SAN :
  - ***NexentaStor*** (Nexenta Enterprise Edition)
  - ***TrueNAS***
  - ***Open-E***
  - …

- ili OS za opću upotrebu, koji ćemo dodatno konfigurirati prema našim potrebama:
  - Neki Linux koji želimo prilagoditi našim potrebama ili
  - Windows server koji već imamo te ga želimo optimizirati ili prenamjeniti za NAS/SAN sustav
  - ili nešto drugo

Postoje i gotovi samostojeći uređaji koji dolaze zajedno s operacijskim sustavom. Proizvode ih : ***EMC***^{2}, ***IBM***, ***Dell***, ***NetApp*** i drugi (oni nisu tema ove priče).

2. Nakon što smo odabrali operativni sustav za NAS ili SAN poslužitelj, moramo biti svjesni i njegovih zahtjeva za hardverom:

- Koji CPU će zadovoljiti naše potrebe
- Koliko RAM memorije (i koji tip) koristiti
- Koje mrežne kartice: 1Gbps ili 10Gbps, koji modeli (chipovi) i koliko ih je potrebno (jedna, dvije, tri, …)
- Koji mrežni preklopnik (wwitch) odabrati, s kojom verzijom *Firmware*a (*OS*a) i s kojim funkcionalnostima Pogledajte članak “Switching i routing: jučer, danas, sutra”
- Koji RAID kontroler, s koliko pripadajuće RAM memorije, BBU i sl. odabrati
- Koje tvrde diskove odabrati
- Koje RAID polje koristiti (**RAID 0**,**1**,**5**,**10**,…)

3. Nakon planiranja resursa koji će nam trebati u upotrebi budućeg NAS ili SAN sustava, potrebno je revidirati točku 2.

Lošim odabirom bilo koje točke, a posebno diskova i RAID kontrolera, ćemo kasnije u radu doći do problema ili vrlo često gubitka podataka, koje ćemo vrlo skupo platiti ($$$).

A što je SAN sustav

SAN (*Storage Area Network*) sustavi ne nude mrežno dijeljenje datoteka, već nam osiguravaju mrežni pristup Tzv. *Block-based* mediju.

S druge strane, **NAS** sustavi koriste diskove koji su particionirani i formatirani odnosno na kojima se nalazi neki datotečni sustav (pr. NTFS, XFS, ext3/4, …). Tom datotečnom sustavu se potom, preko protokola za mrežno dijeljenje (pr. **NFS, SMB/CIFS, FTP**, …), pristupa preko mreže.

Naime **SAN** sustavi, preko mreže praktično dijele svoje diskove ili polja diskova vidljiva klijentskoj (drugoj) strani kao običan tvrdi disk, odnosno površinu tog diska. Takva *površina* diska je prazna, nema niti particija, niti datotečni sustav i prema tome se ponaša kao stvarna prazna površina (novo kupljenog diska).
Nadalje takav disk se sastoji od blokova podataka kao i bilo koji lokalni ATA, SATA, SAS ili neki drugi disk.
Za više detalja o diskovima pogledajte knjigu **Uvod u Linux i Linux napredno - poglavlje o diskovima i particijama.**
Za ovakvo mrežno dijeljenje praktično *sirove* površine diska, potrebi su mrežni protokoli koji nam osiguravaju ovakav pristup.
Neki od SAN protokola su :

- **Fibre Channel**
- **iSCSI**
- **ATA over Ethernet (AoE)**
- **HyperSCSI**.

Nakon što se klijent preko nekog od gore navedenih protokola spoji (na površinu diska), takav disk se mora prvo particionirati te formatirati, kao da se radi o lokalnom disku odnosno disku spojenom na vaše lokalno računalo.

Pojam ***formatiranje*** se odnosi na proceduru instalacije datotečnog sustava na odabranu particiju diska.

Zbog čega uopće koristiti NAS ili SAN sustave

Zašto bi uopće koristili ovakve sustave ?

- Zbog potrebe za izradom sigurnosnih kopija vaših podataka (Engl. Backup), na centralni mrežni uređaj (obično NAS):
  - koji bi morao (i obično je) biti znatno sigurniji jer u pravilo snima podatke na više diskova istovremeno (obično koristi neko RAID polje)
  - i ima sve ostale komponente kvalitetnije i pouzdanije od "običnog" računala
- Zbog centraliziranog pohranjivanja i upotrebe. Dakle želimo centralizirano mjesto za pohranu :
  - Svih podataka (datoteka) s kojima radimo
  - Virtualnih računala, a što je preduvjet za bilo koji rad u *klaster*u, u slučaju potrebe za redundancijom (otpornosti na kvar pojedinog poslužitelja ili njegovog održavanja odnosno nadogradnje).
- Zbog pohranjivanja sigurnosnih kopija virtualnih računala i drugih podataka (pr. dokumenti, slike, …) i to na dnevnoj, tjednoj, mjesečnoj, polugodišnjoj ili godišnjoj bazi.

I NAS i SAN sustavi osim čiste pohrane podataka imaju (i moraju imati) i cijeli niz drugih naprednih mogućnosti koje su vrlo važne za ovu namjenu odnosno upotrebu.

Koje opcije bi najminimalnije morali imati NAS/SAN sustavi ?

Praćenje:

- performansi sustava
- RAID kontrolera i svih njegovih polja
- svakog pojedinog diska (performanse ali i što je još važnije grešaka u radu)
- servisa/daemona
- mreže i mrežnih komponenti

Napredne opcije:

- Firewall
- naprednu konfiguraciju mreže :
  - VLAN-ovi
  - Agregacija/Bonding - i to nekoliko mogućnosti i protokola
- mogućnost replikacije podataka na sekundarni NAS/SAN sustav

- izradu “snapshota” - u zadanim vremenskim okvirima i ručno a koja je po mogućnosti vidljiva i direktno u operacijskom sustavu klijenata (Pr. kao Windows “Previous Versions”)
- optimizacija svake važnije komponente sustava
- Prava pristupa : kreiranje korisničkih grupa kao i pojedinačnih korisnika, uz mogućnost integracije s Active Directory ili LDAP servisima.

Sve ove *napredne* opcije i parametri vrlo su važni u realnim radnim uvjetima. Naime svaka mreža i IT sustav su specifični i imaju specifične potrebe. Većina IT sustava je osim toga otvorena za mnoge optimizacije (jer ih obično nitko i nije optimizirao, barem ne na profesionalan način). Mnogi će se zapitati da li je ovo stvarno potrebno. U praksi (s kojom sam se i sam susreo) reći ću **DA**.

Kreće se od optimizacije:

- operativnog sustava
- do mrežne razine (uz pretpostavku da imate pravu mrežnu opremu) koja je preduvjet za bilo kakav imalo ozbiljniji IT sustav. Ovdje se radi o mrežnim parametrima
- preko optimizacije mrežnih protokola i protokola za NAS ili SAN sustav
- do optimizacija na razini ispod NAS ili SAN sustava (prema RAID razini)

Kako uopće izgledaju NAS ili SAN sustavi ?

Pogledajte slike par NAS/SAN sustava, nekoliko poznatih proizvođača

Synology NAS DS216 sustav, tvrtke **Synology**, koji omogućava ugradnju do 2 SATA diska u jedno kućište.

Slika je u vlasništvu tvrtke **Synology**

DELL Powervault NX3100 sustav tvrtke **Dell**, koji omogućava ugradnju do 12 SAS ili SATA diskova u jedno kućište.

Slika je u vlasništvu tvrtke **Dell**

HPE MSA 1040 sustav tvrtke **HP**, koji omogućava ugradnju do 24 SAS ili SATA diskova u jedno kućište.

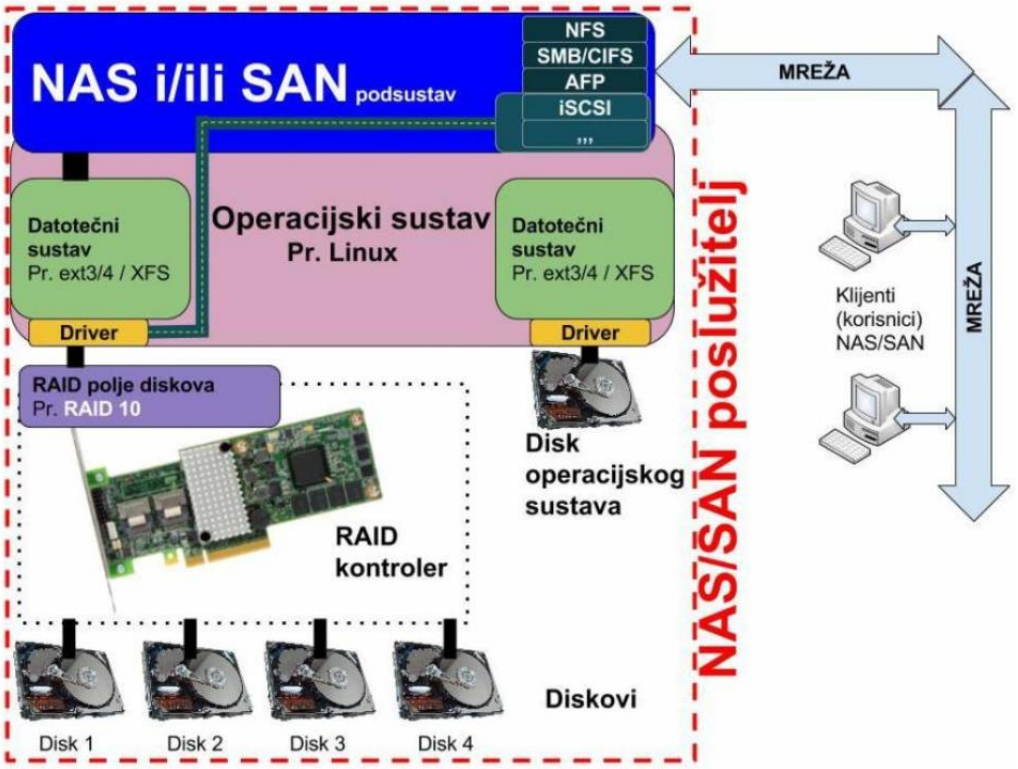
Slika je u vlasništvu tvrtke **HP**

SSG-6048R-E1CR60N sustav, tvrtke **SuperMicro**, koji omogućava ugradnju do 60 SAS ili SATA diskova u jedno kućište.

Slika je u vlasništvu tvrtke **SuperMicro**

Rezimirani pogled na NAS i SAN sustave

Sada ćemo rezimirati sve o čemu smo govorili. Stoga pogledajte logičku shemu NAS i/ili SAN sustava na slici dolje.



- RAID kontroler ima svoj centralni procesor i specijalizirani procesor (obično su obije funkcionalnosti integrirane u jednom *chipu*). Specijalizirani (dio) procesora se zove **RAID** ASIC i on je zadužen za **RAID** polja.
- RAID kontroler ima i svoju **ROM** i **RAM** memoriju kao i disk kontroler na koji se zapravo spajaju diskovi
- RAID kontroler ima svoj *minijaturni* operativni sustav, niže razine a koji je inicijalno pohranjen u **ROM** memoriji i iz koje se učitava. Unutar tog *minijaturnog* operativnog sustava se nalaze metode za pristup svim diskovima spojenim na njega kao i sve potrebno za kreiranje RAID polja unutar kojega će diskovi raditi.

- Promatrajmo diskove koji su u konačnici spojeni na RAID kontroler kao na jednu komponentu koja komunicira s ostatkom računala odnosno operativnim sustavom i to preko upravljačkog programa (Engl. *Driver*) za taj RAID kontroler.
- Operativni sustav preko upravljačkog programa za RAID kontroler vidi samo polja diskova koja su kreirana od strane RAID kontrolera,i to kao jedan jedini disk. Ako smo kao na slici, kreirali **RAID 10** polje unutar kojega se nalaze četiri (4) tvrda diska, operativni sustav (u ovom primjeru je to Linux), vidjeti će jedan jedini tvrdi disk, koji je zapravo cijelo **RAID 10** polje diskova. Taj *logički* disk se s točke operativnog sustava sastoji od svih dijelova od kojih se sastoji bilo koji *normalan* tvrdi disk (trake, cilindri, *klaster*i i sektori)
- Nadalje taj "logički" disk je potrebno *particionirati* te kasnije *formatirati* s nekim *datotečnim sustavom*.
    - Svi programi dalje koriste taj datotečni sustav za pohranjivanje datoteka, što je i slučaj s programima za dijeljenje datoteka preko mreže (NAS) : **NFS**, **SMB/CIFS**, **AFP**, …
    - Što se tiće sustava koji pristupaju podacima na razini blokova (**SAN** sustavi), oni disku pristupaju na razini blokova podataka tj. ispod razine datotečnog sustava.

U čemu je problem sa standardnim NAS ili SAN sustavima

1. Što kada se pokvari ovakav NAS ili SAN sustav odnosno poslužitelj ?
2. Obično ako niste imali još jedan sustav na koji ste izrađivali sigurnosne kopije - ostajete bez svih podataka.
3. Što kada/ako izgubite sve podatke važne za poslovanje vaše tvrtke ?.

Slijedeći korak: Klasterski i/ili redundantni NAS sustavi

Klasterski i/ili redundantni **NAS** sustavi se sastoje od dva ili više **NAS** sustava u paralelnom radu.

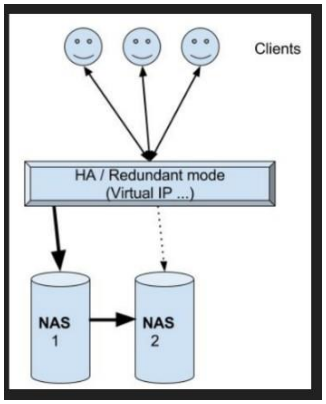
Što nam omogućavaju ovakvi sustavi ?

Osim pouzdanosti, jer se sada svi podaci mogu zapisivati na dva ili više uređaja istovremeno, dolazimo i do njihovih ograničavajućih faktora, a to su :

- Omogućavaju horizontalno skaliranje (nadogradnju) ali uz više troškove - znatno više od cijene samog drugog uređaja
- Ograničeni su na proširenje prostora tj. kapaciteta (proširenje dodavanjem diskova ili dodatnih uređaja), Pri tome proširenje dodavanjem dodatnih uređaja (ako je to uopće moguće jer za svaki model odnosno seriju redundantnih NAS uređaja postoji ograničenje do koliko se mogu proširivati) - cijene ovdje odlaze u nebo.
- U konačnici daju nam redundanciju (zalihost/sigurnost od gubitka podataka) uz ekstra cijenu.
- Uz što veću cijenu, dobivamo i veću brzinu rada
- To su sve uglavnom rješenja koja su zaštićena i zatvorenog dizajna od strane proizvođača poput (EMC, IBM, …)

Redundantni ili Klasterizirani NAS Sustavi

Redundantni NAS Sustavi



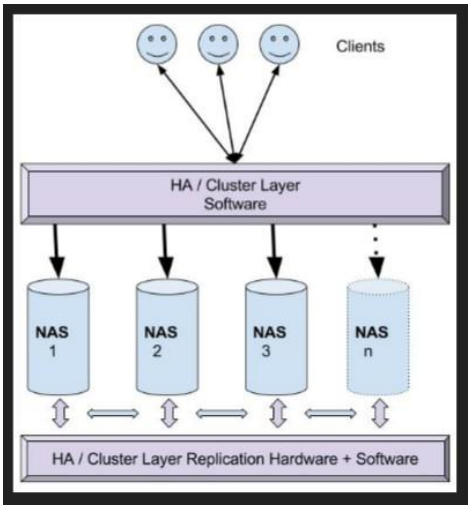
Redundantni sustavi su nešto jednostavnijeg dizajna jer se ispred njih logički nalazi sustav koji osigurava pristup jednoj jedinoj virtualnoj IP adresi kojoj klijenti i pristupaju (postoje i drugačije implementacije ali ova je najčešća).

U pozadini se redundantni NAS sustav brine da se svi podaci uredno kopiraju s prvog (NAS-1) na drugi (NAS-2) NAS sustav.

U slučaju kvara prvog (NAS-1) sustava, drugi (NAS-2) preuzima njegovu funkciju i svi podaci su sačuvani.

Ovakvi sustavi su često izvedeni sa samo dva NAS sustava i rade po principu **: Active-Standby** (jedan je aktivan a drugi je pričuva).

Klasterizirani NAS Sustavi



NAS sustavi koji se nalaze u *klasteru* (grozd) su obično znatno kompleksnijeg dizajna koji uključuje i razne dodatne hardverske i softverske komponente.

Svi klijenti u pravilu pristupaju vršnoj *klasterskoj* komponenti, koja je često izvedena samo u softveru a koja se brine za raspodjelu podataka unutar *klastera*, na pojedine NAS uređaje.

Sama replikacija tih (odnosno svih) podataka između pojedinih NAS sustava se često izvodi i u softveru i na specijaliziranom hardveru (na slici bi to odgovaralo donjem sloju).

Zbog ovakvog, složenog dizajna i potrebe za posebnim hardverom i njihova cijena je poprilično veća od redundantnih NAS sustava.

Softverska rješenja

Postoje i mnoga *Open Source* softverska rješenja koja nam omogućavaju osnovnu redundanciju ili klasterizirane NAS sustave.

Neki od njih su :

- **GlusterFS** : Omogućava osnovne i nekoliko naprednih razina redundancije :
  - mirror - poput RAID 1 između dva NAS sustava (poslužitelja) - min. 2 poslužitelja,
  - stripe - poput RAID 0 između dva NAS sustava (poslužitelja) - min. 2 poslužitelja,
  - mirror + stripe - poput RAID 10 između dva para NAS sustava (poslužitelja) - min. 4 poslužitelja (1 i 2 u RAID 1, 3 i 4 u RAID 1 , oba para poslužitelja (1,2 + 3,4), vršno u RAID 0, što zajedno čini RAID 10)
- **pNFS** (Parallel NFS - od verzije NFS v.4.1+) :Paralelni/Distribuirani NFS - stabilna (produkcijska verzija) je još u izradi
- **OCFS2** (Oracle open source) : ima slične mogućnosti kao **GlusterFS**
- …

Svaki od njih ima svoje prednosti i mane kao i ciljanu upotrebu (za koju je i razvijan ili se pokazao kao vrlo dobar)

Redundantni ili Klasterizirani SAN Sustavi

Slično kao i za redundantne ili klasterizirane NAS sustave - osim sigurnosti jer se sada svi podaci mogu zapisivati na dva ili više uređaja istovremeno, ovdje imamo sljedeće mogućnosti odnosno ograničenja :

- Omogućavaju veće horizontalno skaliranje (nadogradnju) ali uz znatno više troškove - znatno više od cijena za NAS sustave.
- Ograničeni su na ekspanziju prostora tj. kapaciteta (proširenje dodavanjem diskova ili dodatnih uređaja), Pri tome proširenje dodavanjem dodatnih uređaja (ako je to uopće moguće jer za svaki model odnosno seriju redundantnih NAS uređaja postoji ograničenje do koliko se mogu proširivati) - cijene ovdje odlaze u nebo.
- U konačnici daju nam redundanciju (zalihost/sigurnost od gubitka podataka) uz ekstremno visoku cijenu i vrlo kompleksan dizajn.
- u pravilu uz što veću cijenu, dobivamo i veću brzinu rada
- To su sve uglavnom rješenja koja su zaštićena i zatvorenog dizajna od strane proizvođača poput (**EMC²**, **IBM**, …)

Softverska rješenja za SAN i klasterske SAN sustave

I u ovoj kategoriji imamo nekoliko opensource rješenja koje možete proučiti,a koja su dosta česta u upotrebi - i to obično u kombinaciji s drugim elementima odnosno komponentama (ovisno da li se radi o SAN ili klasterskom SAN rješenju).

- **DRBD8** (Distribuirani - replicirani "Block Device") - "Distributed Replicated Block Device" : - praktično RAID1 (mirror) preko mreže, prema principu : Primari poslužitelj → Sekundarni poslužitelj. Potrebna su dva poslužitelja (nije za sve primjene !)
  - **DRBD9** (u aktivnom razvoju) : omogućava rad s više poslužitelja, višestruke replikacije i sl.
- **iscsid (open-iscsi)** (iSCSI initiator) servis/daemon za Linux (sam po sebi nije redundantan već pruža osnovnu iSCSI funkcionalnost)
  - **device-mapper-multipath** - DM-Multipath ("Device Mapper Multipathing") servis/daemon koji omogućava redundanciju (ili load balancing) prema iSCSI uređajima (SAN storage-ima) (i dalje je pitanje kako sinkronizirati dva ili više SAN storage-a) - koristi se najčešće kod active/passive SAN sustava
  - **ALUA** ("Asymmetric Logical Unit Assignment") nudi load balancing prema SAN storage-u (ostaje isto pitanje sinkronizacije SAN storage-a) - koristi se za active/active SAN sustave

ZFS - negdje između

Ali prvo malo o ZFS-u i tvrtki Sun Microsystems

ZFS je razvila tvrtka **SUN Microsystems**, danas u vlasništvu tvrtke **Oracle**. Ideja je bila riješiti gore navedene probleme, uvesti mnoga poboljšanja i mogućnosti koje su do tada bile dostupne samo kao specijalizirana rješenja ili uopće nisu postojala, te sve integrirati u jednom "proizvodu".

ZFS je prema svojoj funkcionalnosti praktično kombinacija:

- Naprednog RAID kontrolera odnosno *Logical Volume Manager*a i
- Datotečnog sustava s naprednim sustavom kontrole (ACL) odnosno *Access* listama za prava pristupa

Izvorno je bio razvijan unutar tvrtke **SUN Microsystems** kao zatvoreni kod, unutar njihovog UNIX operativnog sustava **Solaris** 2005. godine. Već slijedeće godine je prebačen u open source pod CDDL licencom unutar projekta "OpenSolaris" te je postao sastavni dio Solaris UNIX-a v.10, sredinom 2006. godine.

Naravno, ubrzo nakon što ih je kupio **Oracle**, cijeli izvorni kod ZFS-a se više nije održavao od strane Oracle-a. Dakle Oracle je prestao s razvojem *OpenSolaris*a pa je zajednica morala sav kod prebaciti u novi projekt imena *Illumos* (tu se nalazio i kod *ZFS*a). Zajednica koja je stajala iza projekta *Illumos* preuzela je zadnju verziju dostupnog koda te ga nastavila razvijati. Nakon nekog vremena je pokrenut i projekt *OpenZFS* koji je prihvatila još veća zajednica programera i korisnika ili i sve veći broj tvrtki. Svi zajedno su nastavili s razvojem open source verzije *ZFS*a, koja se razvija i danas.

Kao i većina programa ili sustava koji su izašli iz tvrtke **SUN Microsystems** *ZFS* je razvijan od strane inženjera za inženjere, na najbolji mogući način, kao nedostižni uzor svim ostalim tvrtkama (barem za 99.9999% njih).

Borba s *open source* licencama

Pošto je *ZFS* razvijan pod CDDL licencom a koja nije kompatibilna s Linux GPL licencom pod kojom je razvijan Linux kernel, već od početka javnog razvoja (krajem 2005.g i početkom 2006.g.) bilo je jasno da se *ZFS* ne smije direktno implementirati u Linux kernel.

Za linux je osmišljeno privremeno rješenje : upotreba preko FUSE frameworka, unutar kojega su se smjeli pokretati programi s drugim licencama. Problem je bio u tome što se FUSE izvršava na višoj razini iznad kernela te je samim time znatno sporiji. Ali i ovo je bilo nešto za početak. Istovremeno s ovom borbom krenulo se u razvoj ZFS-a od nule te je 2013.g. razvijena prva stabilna verzija (v.0.6.1) - iste godine je pokrenut i projekt "OpenZFS". Godine 2016 s Ubuntu Linuxom v.16.0.4, ZFS koji se razvijao u projektu *OpenZFS* je uključen u ovu distribuciju Linuxa.

Što se tiće drugih open source UNIX operacijskih sustava poput onih koji su razvijani s BSD licencom : **FreeBSD**, **NetBSD**, **OpenBSD** i drugih ovdje nije bilo problema s korištenjem te je ZFS na njima zaživio vrlo brzo te se smatra (zbog godina korištenja i testiranja/popravljanja) kao jedna od najboljih implementacija u open source operacijskim sustavima.

OpenZFS projekt nudi implementaciju *ZFS*a za Mac OS X.

ZFS je nastao u želji da se riješe problemi koje niti najnapredniji **RAID** kontroleri nisu mogli riješiti. Osim rješenja problema, željele su se dodati i neke napredne mogućnosti, koje su većini korisnika bile poželjne i dobrodošle.

Neki od problema koji su poznati a ZFS ih rješava :

- problemi s **RAID 5** i **RAID 6** poljima (pogledajte WIKI)
- problem kada želimo zamijeniti neispravni tvrdi disk s novim diskom (koji na većini RAID kontrolera **mora** biti identičan onome koji se mijenja [pr. po broju glava/cilindara/sektora])
- problemi odnosno komplikacije kod proširenja RAID polja ovisno o RAID polju
- problem u slučaju da nam se RAID kontroler pokvario te ga moramo zamijeniti s novim (ovo je ponekad nemoguće jer možemo izgubiti sve podatke)
- problem kada nam se pokvari pr. matična ploča, te moramo prebaciti sve diskove i RAID kontroler na novi hardver (ovo često može poći po zlu)

- problem koji nastaje s vremenom - podaci na površini tvrdih diskova postaju nekonzistentni a RAID kontroleri nisu toga svjesni, sve dok ne naiđu na problematični dio površine diska. Ovaj problem se naziva *Data decay* ili *Data rot*. On je znan i kao degradacija podataka na površini diska, a tek najnapredniji RAID kontroleri imaju mogućnost (Engl. disk-scrubbing) korekcije ovakvih grešaka i to samo do neke granice. Sličan problem nastaje i uslijed grešaka u firmware-u diska ili RAID kontrolera, *fantomskog* zapisivanja (ako podatak nije stvarno zapisan na površinu diska) ili grešaka kod zapisivanja ili čitanja zbog pristupa prema ili od krivih blokova na ili sa površine diska.

Dodatne Funkcionalnosti ZFS-a :

- komprimiranje podataka "u letu", prema konfigurabilnom algoritmu komprimiranja i razini kompresije. S obzirom na dostupne algoritme za ovu namjenu i brzine ali i mogućnosti modernih *CPU*a, komprimiranje i dekomprimiranje podataka "u letu" je gotovo neprimjetno. ZFS trenutno podržava : LZJB, LZ4, ZLE i GZIP.
- ZFS je i Tzv. *Copy On Write* i *transakcijski* datotečni sustav, što znači da se operacije snimanja rade transakcijski (poput transakcija u SQL bazama podataka). To znači da se svaka operacija zapisivanja završava tek kada je potvrđeno uredno zapisivanje (kada je transakcija uredno završila). Performanse su i dalje zadržane naprednim modelom transakcija te uvođenjem posebnog ZIL log-a za operacije snimanja. Ovaj "Copy On Write" model uvodi i :
    - mogućnost izrade Tzv. *Snapshota* odnosno snimke stanja diska/podataka u vremenu te mogućnost povratka na bilo koji trenutak kada je izrađen bilo koji od *Snapshot*.
    - mogućnost izrade "klonova" odnosno inačice *snapshota* na kojoj se može i zapisivati
- mogućnost naprednih ACL-a (sigurnosnih postavki/prava na datotečni sustav ali i na NFS share direktno)
- … te cijeli niz drugih funkcionalnosti

ZFS u radu

Nakon što smo instalirali ZFS, na njega dodajemo diskove i kreiramo ekvivalente RAID poljima, slično kao što ih dodajemo u neki hardverski RAID kontroler (što se konfiguracije tiće). Tako je moguće kreirati ekvivalente gotovo svim RAID poljima :

- RAID 0 (ovdje se naziva **stripe**) ,
- RAID 1 (ovdje se naziva **mirror**),
- RAID 5 (ovdje se naziva **RAID-Z** ili **RAID-Z1**),
- RAID 6 (ovdje se naziva **RAID-Z2**),
- Nešto poput RAID "6" ali s tri paritetna diska umjesto dva - naziva se **RAID-Z3**
- RAID 10
- …

S jedinom razlikom da to sve radi bez grešaka i problema koje možemo imati na bilo kojem RAID kontroleru a pogotovo ako nismo prethodno testirali sve scenarije u slučaju nekog kvara. Osim toga na današnjem hardveru to sve radi ekstremno brzo a sve je moguće i dodatno drastično ubrzati uvođenjem :

- **L2ARC**-a za ubrzavanje operacija ćitanja (read) i /ili
- **ZIL log**-a za ubrzavanje operacija pisanja (write)

S time da se za obije metode (**L2ARC** i **ZIL log**) mogu koristiti zasebni SSD diskovi a koji dodatno mogu biti i u nekom ekvivalentu RAID polja, da bi se dodatno dobilo na brzini i/ili pouzdanosti.

Pošto je jasno da je ZFS povezan s NFS-om a i vrlo jednostavno sa SMB/CIFS ili nekim drugim sustavom za dijeljenje datoteka (NAS sustavom) vidljiva je njegova upotreba kao naprednog NAS sustava.

Jedna od naprednih stvari oko ZFS-a je i u tome što se na bilo kojem ZFS polju diskova (ekvivalent RAID polju) može kreirati poseban "Block device" koji se može koristiti kao iSCSI logički uređaj (disk). Taj logički disk je samo potrebno proslijediti nekom od iSCSI "serverskih" servisa/daemona. Ovime dobivamo upotrebu ZFS-a kao SAN sustava.

Potencijalna mana upotrebe *ZFS*a leži u tome što nije trivijalan kao što je više manje korištenje RAID kontrolera i kreiranje nekog RAID polja. U svakom slučaju potrebna su vam neka naprednija predznanja. *ZFS*ove stotine opcija, parametara i funkcionalnosti, početnicima mogu izgledati komplicirane ali su profesionalcima definitivno vrlo važne.

Na kraju krajeva niti ne želite da vam NAS ili SAN sustav "složi" netko onako usput, za sat-dva, uz svoj svakodnevni posao koji obično nema veze s ovom temom, jer bi mogli zažaliti kada nešto krene po zlu.

Namjerno nisam napisao "ako" već "kada" jer je uvijek pitanje vremena kada će doći do problema i da li ćete ih biti u stanju riješiti te koliko vremena i novaca će vam biti potrebno za tu "igru".

O ***ZFS*u** su napisane knjige i knjige te više nećemo ulaziti dublje u ovaj najbolji *Volume Manager* i datotečni sustav (svih vremena).

Proces učenja

Ako tek krećete s učenjem, prvo si dajte nekoliko dana da bi dobro savladali:

- osnove Storage tehnologija te napredne mogućnosti
- osnove rada RAID kontrolera te njihove mogućnosti, način rada i dodatne opcije (uz njihovo razumjevanje)

I na kraju dodajte još koji dan za proučavanje foruma koji se bave ovom tematikom, kao i foruma od strane proizvođača s pitanjima i odgovorima vezanim za pojedine (konkretne) modele RAID kontrolera koji vam je ušao u uži izbor (ako ste odlučili da čete koristiti RAID kontroler a ne ZFS).

Kada završite prvu fazu učenja i nakon što ste kupili RAID kontroler, slijedi novo učenje :

- proučite napredne parametre i testirajte ih (istovremeno testirajte i performanse sustava ovisno o promjeni parametara)
- testirajte razne scenarije havarija za barem nekoliko RAID polja (pogledajte dolje za ZFS) i povrata podataka , mjerite i vrijeme koje potrebno da se sve vrati na "staro stanje" - i vrijeme potrebno za "recovery" je ponakad ključno za konačan odabir vrste RAID polja (RAID 1, RAID 5, RAID 6, RAID 10, …)
- sve dobro i detaljno dokumentirajte (ovo ćete najviše cijeniti kad vam se dogodi prva veća greška - višestruko će se isplatiti dani i dani testiranja i dokumentiranja)

Ako ste krenuli prema ZFS-u tada krenite s proučavanjem osnova:

- WIKI ZFS info

Proučite si i pripremite upute iz nekoliko izvora te odvojite još par tjedana za upoznavanje i isprobavanje te smišljajte razne scenarije havarija (i smislite/pronađite najbolje riješenje) :

- Kreirajte jedno ZFS polje:
  - a. Mirror (Ekvivalent RAID 1)
    - 0. Kreirajte share (NFS ili SMB/CIFS), dodjelite ovlasti i dobro ih naučite (proučite kako se dodjeljuju, nasljeđuju, ne nasljeđuju i sl.), Zapišite određene podatke i pratite performanse kod zapisivanja i čitanja (s različitim parametrima).
    - 1. Izvadite jedan disk iz ZFS polja, pa ga vratite
    - 2. Ponovite 1. korak ali prije vračanja diska, obrišite ga
    - 3. Zamijenite mjesta diskovima (prvi na mjesto drugog i sl.)
    - 4. Izvadite sve diskove, reinstalirajte cijelo računalo pa vratite diskove i pokušajte importirati staro ZFS polje
- b. Obrišite postojeće ZFS polje i kreirajte novo (Pr. RAID-Z)
  - Ponovite sve točke : 0 - 4
- c. Obrišite postojeće ZFS polje i kreirajte novo (Pr. RAID-Z2)
  - Ponovite sve točke : 0 - 4
- d. Obrišite postojeće ZFS polje i kreirajte novo (Pr. RAID-10 : 2 x "mirror" u "stripe")
  - Ponovite sve točke : 0 - 4

Napravite što više scenarija te:

- sve isprobajte (testirajte i performanse) i dokumentirajte
- isprobavajte rad s osnovnim postavkama, pa sve probajte optimizirati (za svaki scenarij) - testirate i stabilnost i izdržljivost ovisno o scenariju i opcijama koje ste mijenjali - uz:
  - restart poslužitelja
  - restart servisa/daemona
  - namjerno stopirane servise/daemone (uz ručno pokretanje - naknadno)

U ovim koracima/scenarijima ćete naučiti najviše, te se približiti produkcijskoj primjeni i znanju o ovim sustavima.

Što je slijedeće

Problemi klasterskih NAS i SAN sustava

Kao što smo vidjeli u prethodnom poglavlju klasterski NAS i SAN sustavi imaju svoje limitirajuće faktore. Kod većine je to cijena ali i ograničenja skalabilnosti. Naime veći sustavi često trebaju sve veći i veći kapacitet pohrane podataka, koji postaje ili preskup u startu ili zahtjeva vrlo velika ulaganja kod proširenja. I na kraju krajeva svi oni opet imaju svoje limite, najviše sa strane proširenja.

Kod najvećih igrača poput "Cloud" providera pružanje usluge pohrane velike količine podataka pr. za spremanje virtualnih računala i sl. je svakodnevni posao. Proširivost ovakvih sustava je krucijalna.

Rani odgovor na ovu problematiku je bio razvoj (i kasnija upotreba) sustava koji uopće ne rade na način na koji rade tradicionalni klasterski NAS ili SAN sustavi.

Object storage

I rodio se *Object storage*, koji podatke *promatra* i pohranjuje kao objekte, za razliku od tradicionalnih sustava kod kojih postoji neka struktura datoteka i direktorija (odnosno klasičan datotečni sustav) kod NAS sustava. Ovo je drugačije i od SAN sustava koji rade s blokovima podataka koji se spremaju u sektore na disku (logičkom ili fizičkom).
Kao što RAID kontroler *razlama* neku datoteku na male blokove podataka koje dalje raspoređuje na diskove, ovisno o RAID polju, tako i ovi sustavi "razlamaju" podatke na Tzv. objekte (uz pripadajuće metapodatke), koje onda raspoređuju na poslužitelje u klasteru.
Objektni *storage* trebao bi nam nuditi, skalabilni (proširivi) sustav otporan na greške. Ovakvi sustavi su se počeli znatnije razvijati od 1995 godine iako su neki radovi i ideje nastali i znatno ranije.
Prvo komercijalno riješenje je razvila tvrtka **Centera Technology** koju je odmah kupila tvrtka **EMC²** te je 2002 izbacila na tržište pod tržišnim nazivom "**EMC Centera**". Ova linija proizvoda se i danas razvija.
Smatra se da se u razvoj ove tehnologije od strane neovisnih investitora u prvim godinama uložilo oko 300 milijuna dolara (ova cifra je rasla sve više). Ne računajući ulaganja tvrtki poput : **DataDirect Networks**, **Centera**, **Atmos**, **HDS**, **EMC²**, **HP**, **IBM**, **NetApp**, **Redhat** i drugih a kasnije i od strane *Cloud providera* poput : **Amazon AWS**, **Microsoft (Microsoft Azure)**, **Google (Google Cloud Storage)** i drugih.

Pogledajmo listu nekoliko visoko skalabilnih, redundantnih *Object storage* sustava dostupnih pod nekom od *open source* licenci:

- CEPH (info)
- Lustre (info)
- LizardFS (info)
- Hadoop Distributed File System(info)
- Moose File System (info)
- Quantcast File System (info)
- …

Kod većih sustava, kao i kod sustava kod kojih korisnici NE žele kupovati super skupi hardver i softver za *Object Storage* sustave, jedno od open source rješenja je **CEPH** o kojemu ćemo govoriti dalje u tekstu.

CEPH

Ceph je distribuirani objektni sustav za pohranu podataka (Engl. Storage) koji je dizajniran za postizanje odličnih performansi, te sustav koji je visoko dostupan i pouzdan. Osim toga on je krajnje skalabilan odnosno proširiv do razine Exabyte-a.

Ovo je sustav koji je zbog svog dizajna otporan na greške i kvarove cijelih poslužitelja i/ili pojedinačnih diskova ili grupe diskova, a u većim implementacijama, cijelih ormara punih poslužitelja pa čak i cijelih podatkovnih centara a samim time i desetcima, stotinama ili tisućama diskova. Sve ovisno o konfiguraciji i raspoloživoj opremi.

Više informacije možete pronaći na : http://ceph.com

Malo o povijesti CEPH-a

Razvio ga je Sage Weil kao temu za doktorski rad na sveučilištu *University of California, Santa Cruz*.
Razvoj se nastavio u tvrtki **Inktank**. Navedenu tvrtku je kupio **RedHat** , 30.04.2014 (za 175 milijuna U$ u gotovini). Tvrtka **Red Hat** ga nastavlja razvijati do danas (kao i zajednica koja ga koristi). Projekt je i dalje, i ostati će *open source*.
Da li postoji i podrška od strane proizvođača hardvera
Naravno, vrlo brzo nakon učlanjenja u obitelj **Red Hat** svi važniji proizvođači hardvera počeli su nuditi sustave koji su certificirani za *CEPH*, pr. :

- Supermicro
- HP
- DELL
- … i mnogi drugi

Osim navedenog hardvera, CEPH se može koristiti i na bilo kojem hardveru koji imate a na kojem se može pokretati bilo koja **RedHat** ili **Debian** bazirana distribucija Linuxa, imalo novije generacije. Dakle dostupni su RPM i Debian paketi.
Osim toga dostupan je i izvorni kod CEPH-a, pa je sve moguće kompajlirati i za druge distibucije Linuxa.

Integracija

CEPH klijent se već standardno nalazi unutar Linux kernela. Server je dostupan ionako kao open source na stranici : http://ceph.com/resources/downloads/.

Osim navedenog CEPH je trenutno integriran s dvije platforme za virtualizaciju:

- **Open Stack** - info :
  - Integriran je sa : **Nova**, **Cinder** i **Glance** za *Block storage*
  - Integriran je sa **Keystone** i **Swift** za *Object storage*
- **Proxmox VE** - pogledajte info :
  - Kao *Block storage* za virtualna računala i za Linux kontejnere

Tko ga trenutno koristi

Koriste ga i najveći igrači, poput :

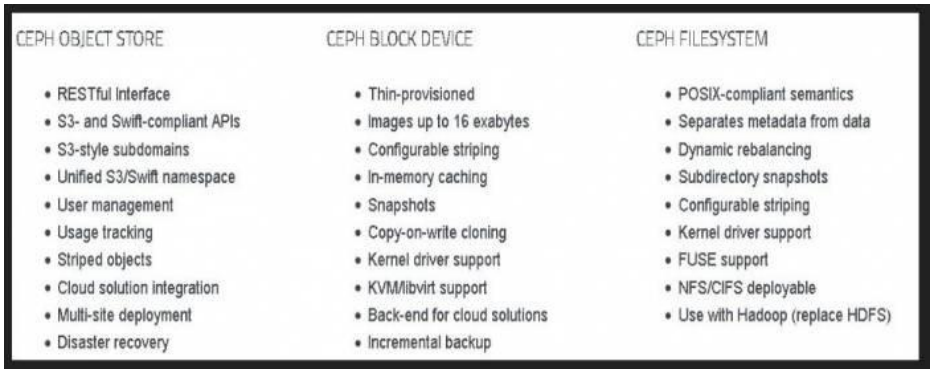
- **Amazon AWS** - prema nekim informacijama, koristi se za neke dijelove S3 Storage sustava
- **Facebook** - za neke dijelove sustava
- **CERN** - prema podacima od prošle godine - koriste ga za ukupno 1+ PB (za spremanje podataka)
- **DreamHost** (Web hosting provider) :
  - 2+ PB za S3
  - 3+ PB kao *Block Device* - za virtualke
- … i mnogi drugi (mnogi i ne žele iznositi što točno koriste iz sigurnosnih razloga)

Za što se sve može koristiti CEPH

CEPH iako radi s objektima na najnižoj razini, na vršnoj se može koristiti za tri različite "upotrebe", i to :

- Kao ***Block Device*** i to ako se koristi kao *Rados Block Device* (RBD) - vidljiv dalje kao *Block Device* ili logički disk koji se koristi za opću upotrebu (pr. za spremanje diskova virtualki i sl.)
- Kao ***Object Storage*** preko *RADOSGW*a, a koji je **S3** i **Swift** kompatibilan - najčešće se koristi za snimanje/čitanje datoteka bilo kojeg tipa preko web-a (korištenjem "put" ili "get" metoda)
- Kao ***Filesystem*** tj. direktno kao datotečni sustav, preko *CEPHFS* - može se *mount*ati kao običan datotečni sustav

Pogledajte i malo više detalja :



Odabirom pojedinog modela :

- *CEPH Block Device*
- *CEPH Object Stoage* ili
- *CEPH Filesystem*

moramo koristiti i dodatne servise odnosno funkcionalnosti koje su nužne za ovakav rad. Prema tome potrebno je detaljnije se upoznati sa zahtjevima i načinom implementacije te konfiguracije svakoga od njih.

Prednosti CEPH-a

Osnovne prednosti CEPH-a (i u kombinaciji s Proxmox VE platformom za virtualizaciju) su :

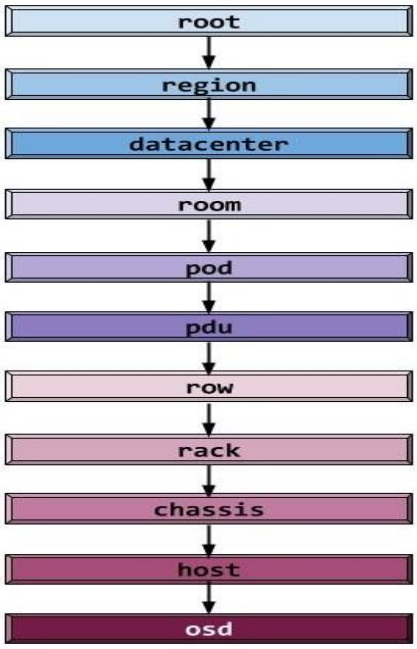
- (Relativno) Jednostavan setup i management iz naredbene linije i grafičkog sučelja Proxmox VE
- "Thin provisioning" (minimalno zauzeće stvarnog diskovnog prostora s podacima)
- Izrada Snapshot-a podataka (datoteka) u letu (dok se radi na njima)
- Automatsko popravljanje grešaka u radu (kod ispada diska, poslužitelja i sl.)
- Ne postoji niti jedna komponenta sustava koja nije redundantna (zalihost)
- Sustav je skalabilan (proširiv) do razine Exabyte-a
- Moguća je konfiguracija više segmenata (Engl. Ceph Pools) polja za pohranu podataka, te razina performansi/replikacije za svaki segment
- Svi podaci unutar polja su replicirani, čineći cijelo polje otpornim na kvarove
- CEPH je moguće instalirati i na pristupačan hardver
- Nema potrebe za RAID kontrolerima ("zabranjena" je njihova upotreba - kao i kod ZFS-a (kod kojega je to izričito **ZABRANJENO**))
- CEPH je razvijan kao "open source" prema licenci LGPL 2.1

Kako se podaci distribuiranju unutar cijelog CEPH clustera

Koristi se Tzv. CRUSH algoritam i pripadajuća "CRUSH" tablica (koja je distribuirana na više poslužitelja) a koja je zadužen za distribuciju, replikaciju i redistribuciju podataka unutar CEPH clustera.

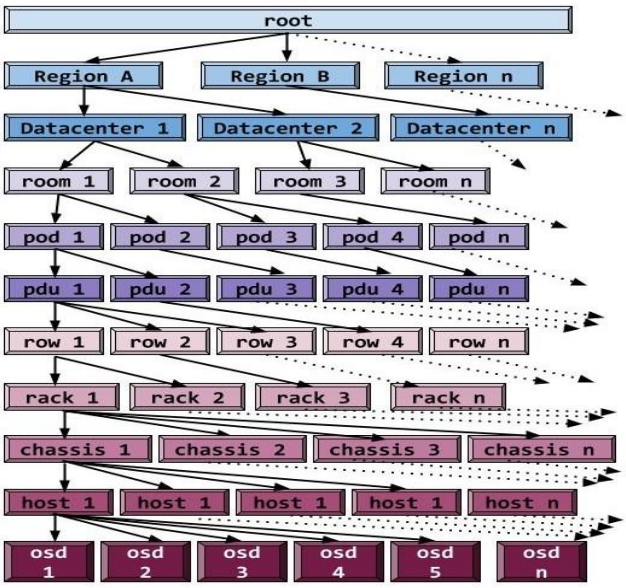
CRUSH je dizajniran da omogućava raznoliku upotrebu, ovisno o veličini implementacije.

Prema tome postoje "CRUSH" tipovi koji opisuju fizičku poziciju CEPH-a unutar cijelog CEPH clustera. Drugim riječima definiramo fizičku hijerarhijsku strukturu svakog elementa unutar hijerarhije (na slijedećoj stranici)



1. **root** - predstavlja vršnu komponentu cijelog CEPH-a - nazovimu ju "cijelom planetom"
2. **region** - predstavlja prvu nižu hijerarhiju - recimo kontinent
3. **datacenter** - predstavlja pojedini podatkovni centar
4. **room** - predstavlja "sobu" unutar podatkovnog centra
5. **pod** -predstavlja logičku podjelu unutar jedne "serverske" sobe - može predstavljati i jedan dio podatkovnog centra koji može biti podjeljen na više ovakvih potencijalno nezavisnih (što se tiče mreže, napajanja, klimatizacije i sl.
6. **pdu** (*Power Distribution Unit*) odnosno podjela prema izvoru napajanja (u podatkovnim centrima ih imamo više pa je ovo dobrodošla dodatna razdioba)
7. **row** - predstavlja jedan red s ormarima punim poslužitelja
8. **rack** - predstavlja jedan ormar s poslužiteljima
9. **chassis** - predstavlja jedno kučište unutar kojega može biti više poslužitelja - misli se na Blade poslužitelje.
10. **host** - predstavlja jedan poslužitelj
11. **osd** predstavlja, u konačnici, pojedinačni disk, unutar poslužitelja

Osim toga u svakoj kategoriji u hijerarhiji, može biti i više elemenata na istoj razini - poput ovoga na slici **dolje**.



Ovakav hijerarhijski model nam omogućava stvarno raznolike scenarije upotrebe.

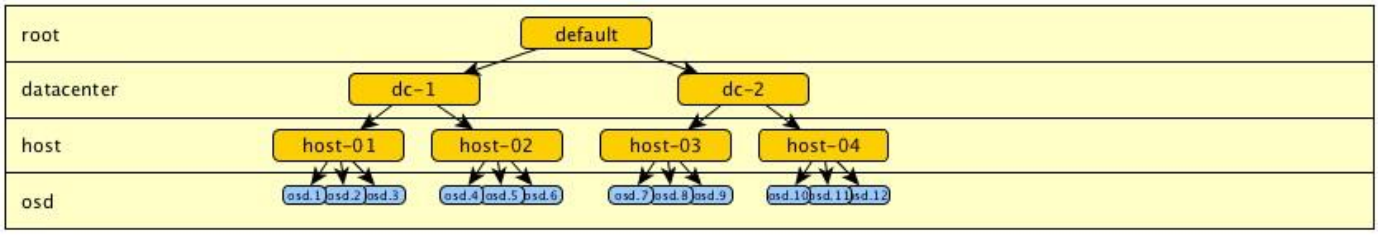
Stoga CEPH može biti implementiran od najmanjih sustava - pr. s minimalno tri (3) poslužitelja s diskovima a s druge strane na sustavima koji imaju tisuće poslužitelja s diskovima, koji su raspoređeni u konačnici na veliki broj podatkovnih centara.

Dakle od CEPH sustava koji imaju nekoliko poslužitelja (svaki s par ili znatno više diskova), preko cijelih ormara (Engl. *Rack*) punih poslužitelja, do *serverskih* soba punih takvih ormara ili cijelih podatkovnih centara.

Dodatno CEPH se uredno može proširiti između više podatkovnih centara pa i više kontinenata na kojima se može nalaziti veliki brojev podatkovnih centara.

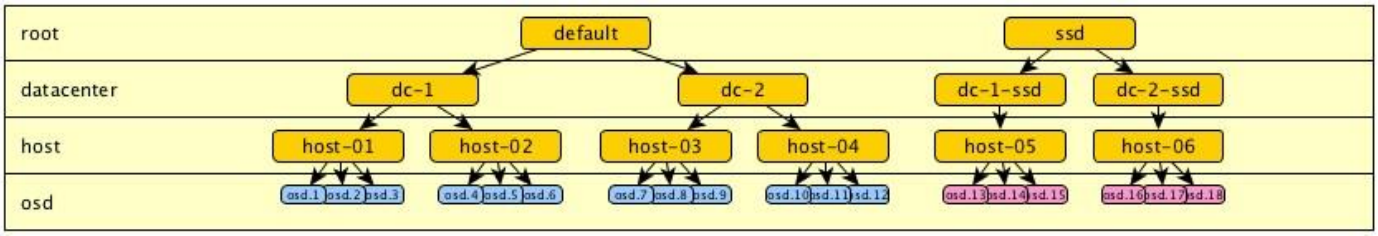
Pogledajmo nekoliko mogućih osnovnih scenarija :

1. Dva podatkovna centra, svaki s par poslužitelja



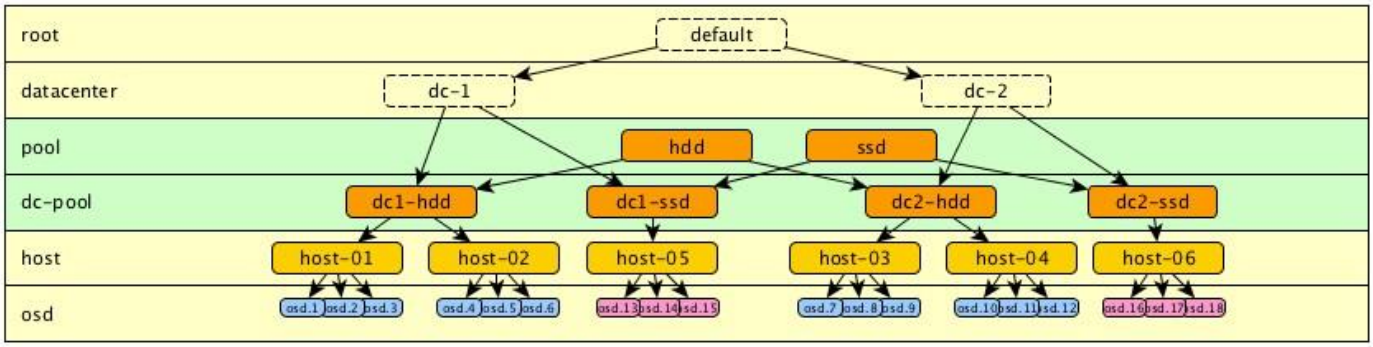
Vidljivo je da unutar svakog podatkovnog centra (**datacenter**) imamo dva poslužitelja (**host**) od kojih svaki ima po tri tvrda diska (**osd**)

2. Prošireni scenarij u kojemu isto imamo dva podatkovna centra ali sada imamo poslužitelje s običnim (tvrdim diskovima) i poslužitelje sa SSD diskovima. Poslužitelji s "običnim" diskovima su u jednoj "grupi" a oni s SSD diskovima u drugoj "grupi".



2.1. Logička shema dolje prikazuje i inicijalizaciju Tzv. ***Pool***a.
U CEPH terminologiji *Pool* je ono što bi u RAID-u bilo RAID polje diskova.
Moguće je imati više *Pool*ova, svaki sa svojom konfiguracijom.
Svaki pojedini **Pool** može biti za svoju namjenu:

- brzina
- pouzdanost
- vrijeme odziva
- georeplikacija
- …



U primjeru na slici u svakom podatkovnom centu imamo poslužitelje sa SSD i poslužitelje s običnim tvrdim diskovima.

- Vršno **Pool** - **hdd** koristi sve poslužitelje koji imaju obične diskove
- Vršno **Pool** - **ssd** koristi sve poslužitelje koji imaju SSD diskove

Kod kreiranja ***Pool***a (to je korak koji možete vidjeti u tekstu o radu CEPH-a) odabiremo koliko replika će imati, kao i druge parametre.

Slike su preuzete sa http://cephnotes.ksperis.com/blog/2015/02/02/crushmap-example-of-a-hierarchical-cluster-map

Kako se zapisuju podaci na CEPH cluster

Nakon što je definirana hijerarhijska struktura za CEPH cluster (CRUSH) te kreiran ekvivalent RAID polja koji se prema CEPH terminologiji naziva ***Pool*** sve je spremno za rad.
Pojednostavljeno svaka datoteka koja se zapisuje, lomi se na manje blokove, koji se onda u konačnici zapisuju odnosno **distribuiraju** na dostupne poslužitelje i njihove diskove.
Dakle ako smo za određeni ***Pool*** na kojem radimo, kod kreiranja odabrali da je broj **Replika** odnosno prema CEPH terminologiji **CEPH *Pool Size*** jednak tri (3), to znači da se podaci zapisuju na odredišni poslužitelj a potom na jod druga dva (2) poslužitelja. Tako da ćemo u ovom slučaju isti podatak imati sveukupno na tri (3) mjesta.
Veličina bloka je standardno 4 MB ali se može promijeniti do razine više MB - ovisno o vrsti podataka koje zapisujemo ili čitamo. To znači da je za neke primjene ova veličina zadovoljavajuća a za neke je ova veličine premalena jer se zapisuju ili čitaju podaci koji zahtijevaju dohvaćanje većih blokova podataka odjednom. Promjenom veličina bloka možemo poboljšati performanse i smanjiti opterećenje sustava - zbog smanjenja broja operacija dohvaćanja velikog broja malih objekata.

Ulazno/izlazne operacije prema diskovnom sustavu kod pisanja ili čitanja se zovu IOPS-i. Klasični (magnetski) odnosno mehanički diskovi su znatnije pogođeni ovim operacijama od SSD diskova. Dakle SSD diskovi u prosjeku mogu podnijeti desetke, stotine i tisuće puta više ulazno/izlaznih operacija u sekundi, od mehaničkih/magnetskih diskova.

Proces distribucije podataka

Podaci se distribuiraju na cijeli CEPH *cluster*, sve njegove poslužitelje i njima dostupne tvrde diskove, te se istovremeno radi replikacija, svakog bloka podataka na drugi poslužitelj odnosno disk na njemu. Sve prema tome kako je konfigurirana hijerarhija za CRUSH te koliko replika smo odabrali za određeno CEPH polje odnosno ***Pool***.

Proces zapisivanja i dodatno replikacije, radi se transakcijski (pogledajte ZFS i transakcijski model) - zbog konzistentnosti podataka.

Kod procesa čitanja se također prema klasterskoj tablici i CRUSH algoritmu zna (određuje/izračunava) koji blok podataka je završio na kojem poslužitelju, i na kojem disku na njemu, te se počinje s čitanjem blokova podataka - sa svih poslužitelja i svih diskova.

U konačnici sve se svodi na to da se podaci zapisuju na sve poslužitelje te se kod čitanja također čitaju sa svih njih. Ovime se znatno povećavaju performanse : što više poslužitelja to je brže zapisivanje ili čitanje.

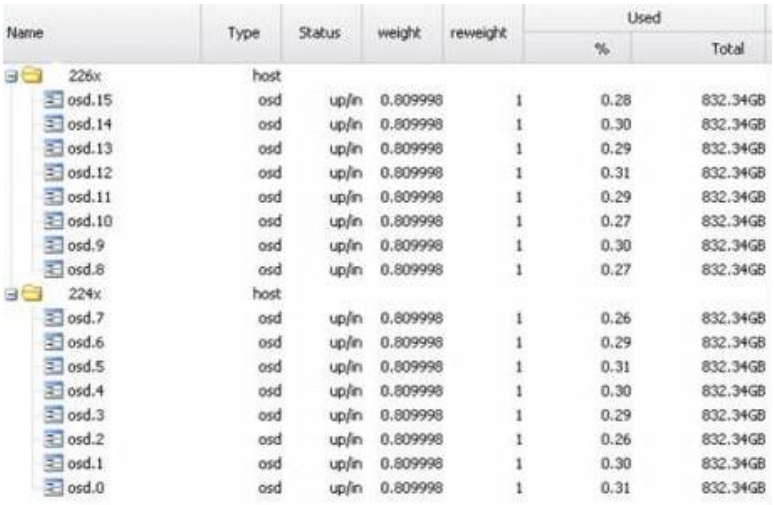
Redistribucija podataka

Što u slučajevima kada se primjerice :

- poslužitelj gasi (zbog kvara, održavanje ili bilo kojeg razloga) ili se dodaje novi
- dodaje se novi poslužitelj
- dodaju se novi diskovi u postojeće poslužitelje ili se neki diskovi vade

Tada CEPH radi Tzv. redistribuciju podataka.

Pogledajte sliku upotrebe CEPH-a na Proxmox VE platformi za virtualizaciju:



Name	Type	Status	weight	reweight	Used	
					%	Total
226x	host					
osd.15	osd	up/in	0.809998	1	0.28	832.34GB
osd.14	osd	up/in	0.809998	1	0.30	832.34GB
osd.13	osd	up/in	0.809998	1	0.29	832.34GB
osd.12	osd	up/in	0.809998	1	0.31	832.34GB
osd.11	osd	up/in	0.809998	1	0.29	832.34GB
osd.10	osd	up/in	0.809998	1	0.27	832.34GB
osd.9	osd	up/in	0.809998	1	0.30	832.34GB
osd.8	osd	up/in	0.809998	1	0.27	832.34GB
224x	host					
osd.7	osd	up/in	0.809998	1	0.26	832.34GB
osd.6	osd	up/in	0.809998	1	0.29	832.34GB
osd.5	osd	up/in	0.809998	1	0.31	832.34GB
osd.4	osd	up/in	0.809998	1	0.30	832.34GB
osd.3	osd	up/in	0.809998	1	0.29	832.34GB
osd.2	osd	up/in	0.809998	1	0.26	832.34GB
osd.1	osd	up/in	0.809998	1	0.30	832.34GB
osd.0	osd	up/in	0.809998	1	0.31	832.34GB

Na slici su vidljiva samo dva poslužitelja **225x** i **224x** (iako su u testu bila tri (i **223x**)) od njih svaki ima po 8 tvrdih diskova (u tablici)

Pogledajte stupac ***Used*** (na slici gore lijevo) i to postotke (kreću se od **0.27** do **0.31**).

Poslužitelj	Diskovi	Kapacitet diskova
225x	OSD.8 OSD.9 OSD.10 OSD.11 OSD.12 OSD.13 OSD.14 OSD.15	(svaki tvrdi disk ima dostupni kapacitet 832GB)
224x	OSD.0 OSD.1 OSD.2 OSD.3 OSD.4 OSD.5 OSD.6 OSD.7	(svaki tvrdi disk ima dostupni kapacitet 832GB)
223x	OSD.16 OSD.17 OSD.18 OSD.19 OSD.20 OSD.21 OSD.22 OSD.23	(svaki tvrdi disk ima dostupni kapacitet 832GB)

Kod dobro balansiranog sustava, postotak zauzeća (upotrebe) svih diskova mora biti podjednak. Za to su zaduženi automatizmi o kojima ćemo malo kasnije.

Dodavanjem novog diska, vađenjem jednog od njih ili dodavanjem/izbacivanjem cijelog poslužitelja sa svim diskovima CEPH kreće u redistribuciju svih podataka. To znači da ako smo recimo dodali novi poslužitelj s osam diskova (detaljnije se radi i o koeficjentu svakog diska ovisno o njegovom kapacitetu i drugim parametrima) podaci se preraspoređuju unutar cijelog klastera i svih diskova, tako da svi diskovi na svim poslužiteljima budu podjednako zauzeti.

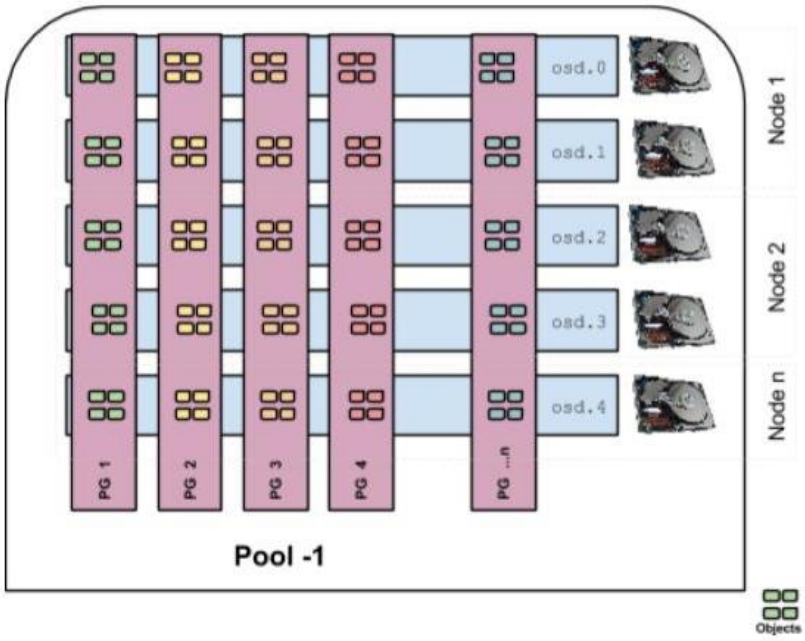
Ovo je vrlo važno jer se nakon dovršetka redistribucije podaci tada počinju zapisivati ili čitati i s tog novog poslužitelja ili novog diska, ravnomjerno koristeći sve resurse (poslužitelje i diskove) klastera.

Za Redistribuciju kao i za replikaciju podataka, koristi se (preporuča) zasebna mreža - da se ne opterećuje “radna” mreža.

Prema CEPH preporukama, potrebno je imati dvije zasebne mreže :

- ***Public Network*** - preko nje čitamo i pišemo podatke na CEPH
- ***Cluster Network*** - preko nje se odrađuju sve ostale radnje poput redistribucije i replikacije podataka

Logička shema cijelog sustava



Logička shema je vidljiva na slici:

- Podaci se spremaju kao objekti
- Objekti se nalaze unutar **Pool**-a
  - Standardna veličina objekta je 4MB
- Objekti se grupiraju u "Placement Grupe" (PG). Placement Grupe su distribuirane preko više OSD-ova (diskova)
- OSD-ovi se koriste za stvarnu distribuciju (*read* i *write* operacija) objekata na tvrde diskove
- **CRUSH** tablica/konfiguracija se koristi za kreiranje i kasniju upotrebu i distribuciju objekata (podataka) unutar svakog pojedinog **Pool**a za cijeli CEPH klaster.

Moguće je imati i više **Pool**ova s različitim konfiguracijama

Pool promatrajte kao RAID polje.

Malo detaljnije

Iako se podaci u konačnici zapisuju kao objekti, odnosno najmanji blok podataka je jedan objekt, standardne veličine 4MB, objekti se prvo grupiraju u Tzv. ***Placement*** grupe. Ove ***Placement*** grupe prema tome povezuju niz objekata koji su dalje raspoređeni na niz ***OSD****ova*.
Pohrana objekata na *OSD*ove znači pohranu na niz tvrdih diskova, raspoređenih na više poslužitelja - ovisno o Pool-u i hijerarhijskoj strukturi definiranoj u *CRUSH* tablici/konfiguraciji.
Prisjetimo se da *CRUSH* tablica/konfiguracija definira fizičku topologiju cijelog CEPH klastera koju koristi CRUSH algoritam za određivanje (izračun) točnih pozicija na koje će se podaci (u konačnici objekti) i njihove replike spremati odnosno čitati.
Sve operacije čitanja i pisanja se zapravo rade na razini svake pojedine *Placement* grupe a ne na razini svakog pojedinog objekta. U protivnom bi rad na razini svakog pojedinog objekta uz dohvaćanje metapodataka za svaki objekt drastično usporilo cijeli sustav.

Placement grupe rješavaju problem s performansama, jer se transakcije događaju na razini *PG*a, kao i pohranjivanje ili baratanje s pripadajućim metapodacima, koje su definirani za cijelu placement grupu (*PG*) a n pojedini objekt u njoj.
CEPH kod čitanja ili pisanja radi na razini *Placement* grupa i njihovih metapodataka (koji ih opisuju), i to transakcijski.
Osim poboljšanja performansi,uvođenjem *Placement* grupa, poboljšala se i skalabilnost (proširivost) cijelog CEPH sustava.
Odnos između broja objekata i broja "Placement" grupa se može okvirno izračunati ili utvrditi testiranjem. Prema preporukama, osnovna računica je :

PG Calculation (formula):

$$PG = \frac{Nr.of\ OSDs\ \cdot\ \mathbf{100}}{Nr.of\ replicas}$$

Za što bolji odabir odnosno izračun broja *Placement grupa* potrebo je uzeti i druge parametre (o tome kasnije).

Pool i PG

Možemo promatrati *Placement* grupe (PG) kao segmente unutar svakog logičkog ***Pool****a* odnosno polja (objekata) na koje se logički spaja svaki CEPH klijent za čitanje ili pisanje na CEPH klaster.

Dakle CEPH vršno gledano, sprema podatke unutar **Pool**a, koji predstavlja logičku grupu *PG*ova. Pool se brine i o tome koliko je primjerice replika potrebno izraditi kod svakog zapisivanje podataka. CEPH može raditi i *snapshot* **Pool**a, u bilo kojem trenutku - kao snimku stanja u vremenu.

CEPH Block Device (Rados Block Device) tj. RBD

Mi ćemo se dalje u tekstu fokusirati na upotrebu CEPH Block devicea.
Prema tome druga dva modela (*CEPH Object Storage* i *CEPH Filesystem*) više nećemo spominjati.

Potrebne funkcionalnosti (CEPH Roles) za RBD

Kao što smo rekli za svaki od CEPH modela, potrebne su određene funkcionalnosti na strani CEPH poslužitelja u CEPH klasteru.
Za upotrebu CEPH-a kao *Block device*a tj. kao *RBD*a, potrebne su nam dvije funkcionalnosti odnosno uloge poslužitelja. To prema definiciji znači da moramo imati poslužitelje od kojih je svaki zadužen samo i isključivo za jednu ulogu:

- uloga Monitor poslužitelja (Engl. *Monitor Node*)
- uloga OSD poslužitelja (ovo su poslužitelji na kojima se nalaze tvrtdi diskove koje ćemo koristiti u CEPH klasteru).

Preporuka za najosnovniju upotrebu kao CEPH RBD, bi bila:

- minimalno 3 poslužitelja s ulogom *Monitor*
- minimalno 3 poslužitelja s ulogom *OSD*
-

Mi ćemo, s obzirom da imamo samo tri poslužitelja s diskovima (koje želimo koristiti kao CEPH kalster za *Block device*) te stoga što što ne tražimo ekstra/turbo brz/proširiv/… sustav, napraviti slijedeće.
Uloge poslužitelja:

- Poslužitelj 1 : **OSD** i **MON**itor
- Poslužitelj 2 : **OSD** i **MON**itor
- Poslužitelj 3 : **OSD** i **MON**itor

Dakle svaki poslužitelj će imati i OSD i MONitor ulogu. S ovime smo na malo zaobilazan način osigurali da imamo i tri OSD-a i tri MONitora.

Zbog čega minimalno tri (3) poslužitelja za klaster

Većina klastera u radu rade na principu "Quoruma" dakle tri je najmanji broj poslužitelja u kojemu minimalna većina (dva) poslužitelja sudjeluju u dogovaranju i provjerama rada.

Ovdje se radi o sustavu glasovanja i izbora što znači da svaki poslužitelj ima jedan glas za glasovanje. Ako su samo dva poslužitelja u sustavu glasovanja izbori su nemogući. Prema tome za sustav glasovanja je potrebno minimalno troje.

Quorum pojednostavljeno

U ovakvim minimalnim klasterima s tri poslužitelja, u svakom trenutku moraju biti aktivna i funkcionalna dva (2) poslužitelja. Ovo ne mora čak značiti da je jedan poslužitelj ugašen već možda ne radi kako treba, pa daje pr. krive rezultate (ili ih ne daje uopće) tada se ta zadnja dva pokušavaju sustavom glasovanja dogovoriti. Ovakav sustav **Quoruma** se koristi i kod klasterskih sustava za virtualizaciju pr. Proxmox VE cluster.

Zamislimo tri poslužitelja koja imaju *Cluster Map* tablicu s pripadajućom verzijom tablice i njen hash/checksum koji govori o tome da li je integritet tablice narušen.

Primjer :

Prva dva poslužitelja kažu da im je zadnja verzija v.234 te HASH : A348F3609D a treći poslužitelj tvrdi da je njegova zadnja verzija v.252 te HASH : 35D56FAB5D. Dogoditi će se to da će prva dva nadglasati treći iako ima veći broj verzije (što bi značilo da je novija) te se on IZBACUJE iz klastera te se više ne uzima u obzir koje slijedeće provjere (sve dok i on ne bude imao sve iste "podatke" kao i preostala dva).
Obično kod ovakvih sustava postoje Tzv. "Izbori" za *klaster Mastera*, a koji se događaju svakih nekoliko sekundi (pr. svakih 15. sekundi).
Dakle u jedinici vremena unutar koje se događaju izbori (ili reizbori) za *Mastera*" tj. *Primarnog* poslužitelja, svaki poslužitelj ima određeni prioritet: Pr. :

- Prvi poslužitelj - prioritet 1
- Drugi poslužitelj - prioritet 2
- Treći poslužitelj - prioritet 3

Ako se recimo onaj s najmanjim brojem prioriteta bira za *Master*a (tj. Primarnog) , tada će "Prvi poslužitelj" postati *Master* ako je sve u redu s njegovim verzijama i integritetom. Ako nije tada će *Master* postati onaj s prioritetom 2 tj. "Drugi poslužitelj" itd. Dakle svakih recimo 15. sekundi se odabire novi *Master*.

Master je obično zadužen za vrlo važne operacije odlučivanja - koji će poslužitelj biti izbačen iz klastera te će on to i fizički napraviti (obično zapisati u datoteku u kojoj je lista aktivnih poslužitelja u klasteru). Ova funkcionalnost je ne zahtjevna prema resursima ali kao što je vidljivo, vrlo važna. "Master" osim toga radi još nekoliko resursno ne zahtjevnih zadaća - ovisno o vrsti i tipu klastera.

Ovo znači da ako primjerice restartamo cijeli klaster (recimo zbog nadogradnji sustava), da to radimo oprezno. Prvo jedan poslužitelj, pa kada je on potpuno funkcionalan nakon restarta, drugi, pa kada je drugi nakon restarta funkcionaln, tek onda treći.

MONitor uloga u CEPH clusteru

MONitor uloga mora biti instalirana na minimalno tri poslužitelja. Ona se brine o:

- tome koji poslužitelji u CEPH klasteru su živi OSD poslužitelji i koji su sve dostupni diskovi (OSD-ovi).
- Pohranjuje i održava 5 tablica odnosno konfiguracija:
    - **Monitor map** - tablica s MONitor poslužiteljima
    - **OSD map** - tablica s OSD poslužiteljima/diskovima
    - **PG map** - tablica s PG (Placement Group)- grupama za pohranu objekata
    - **CRUSH map** - "CRUSH" hijerarhijska tablica/konfiguracija
    - **MDS map** (za **MDS** ulogu [koristi se samo za **S3** ili **Swift** tj. za upotrebu kao "Object Storage"])

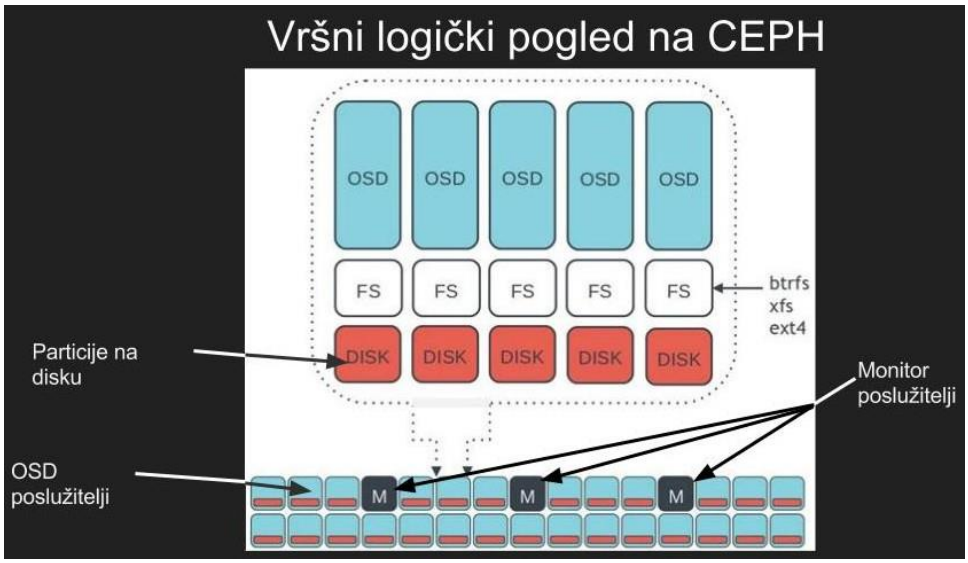
OSD = Object Storage Daemon. Servis (daemon) je to zadužen za rad s objektima i njihovu distribuciju te u konačnici snimanje na tvrdi disk. Jedan OSD daemon (servis) je zadužen za jedan tvrdi disk.
Dakle OSD poslužitelj koji ima osam (8) tvrdih diskova, ima i pokrenuto osam (8) OSD daemona (servisa).

OSD uloga u CEPH clusteru

Ovu ulogu moraju imati minimalno tri (3) poslužitelja.
OSD uloga je zadužena za :

- Spremanje objekata na lokalni datotečni sustav (u konačnici na "OSD" tvrtde diskove) i omogućavanje pristupa objektima preko mreže
- zadužena je za replikaciju objekata koji se zapisuju na konkretni OSD (Daemon/servis) odnosno tvrdi disk. Dakle radi replikaciju objekata koji završe zapisani na OSD (Tvrdi disk) prema drugom OSD (tvrdi disk) - ovisno o *Cluster Map*i i drugim parametrima (tj. o ***Pool***u ili ekvivalentu RAID polja koje se rasprostire na poslužitelje i diskove u CEPH klasteru).
- korištenje journaling mehanizama kod zapisivanja podataka na OSD (disk) prema transakcijskom modelu.

Pogledajmo kako logički izgleda cijeli CEPH, sada kada smo se upoznali sa svim važnijim elementima.



U gornjem dijelu slike je vidljiv izgled jednog **OSD** poslužitelja s pet tvrdih diskova. Svaki tvrdi disk mora imati minimalno jednu particiju, koju možemo formatirati s nekim od predloženih datotečnih sustava:

- **xfs** (preporuka)
- **ext4** ili
- **btfrs**

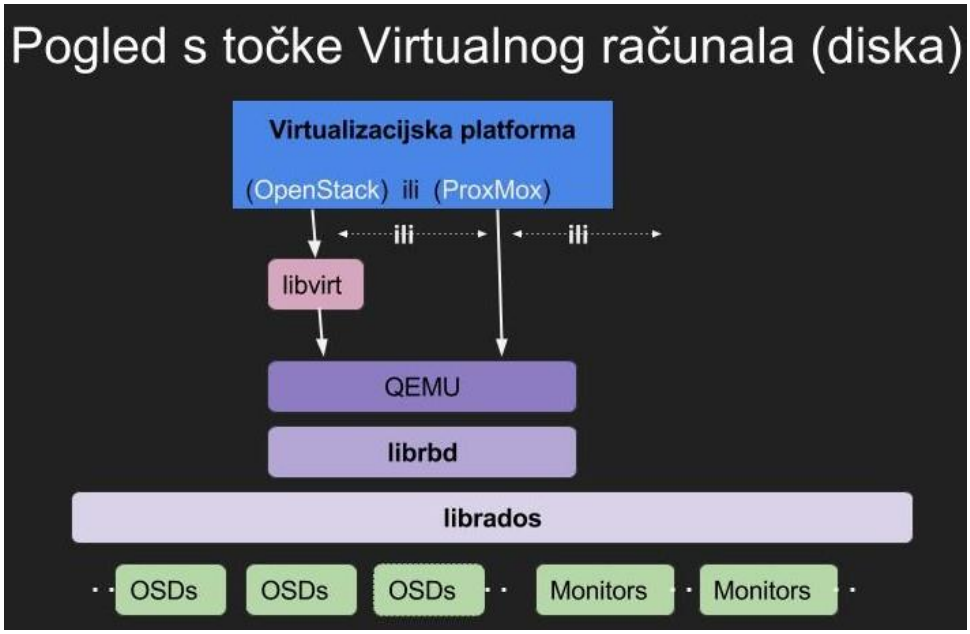
Dodatno, potrebna nam je još jedna particija (ili zaseban disk ili polje diskova s dodatnom particijom za *Journaling*)

U konačnici, na postojeću particiju koja je namjenjena za CEPH, na datotečni sustav kreira se struktura direktorija u koju se spremaju CEPH objekti kao i njihovi pripadajući metapodaci.

U donjem dijelu slike je vidljiva pozicija svakog pojedinog OSD poslužitelja (s svim njegovi *OSD* diskovima) te pozicije svih **MON**itor poslužitelja.
Dakle vidljiv je CEPH sustav sa ukupno 30 poslužitelja i to :

- **3 *CEPH MON***itor poslužitelja i
- **27 *CEPH OSD*** poslužitelja.

Sada zamislimo upotrebu u kojoj imamo poslužitelje za virtualizaciju, koji koriste ovakav CEPH sustav (sa svih 30 poslužitelja) kao disk *storage* sustav, dakle za spremanje virtualnih diskova *virtualki*.



Pogledajmo sliku kako to izgleda sa strane Virtualnog računala odnosno platforme za virtualizaciju prema CEPH sustavu (od gore do dolje)

Ovdje je vidljiv način pristupa CEPH *Block device*u tj. logičkom *blok* uređaju, odnosno disku koji predstavlja cijeli CEPH cluster. Na primjeru su dvije česte platforme za virtualizaciju:

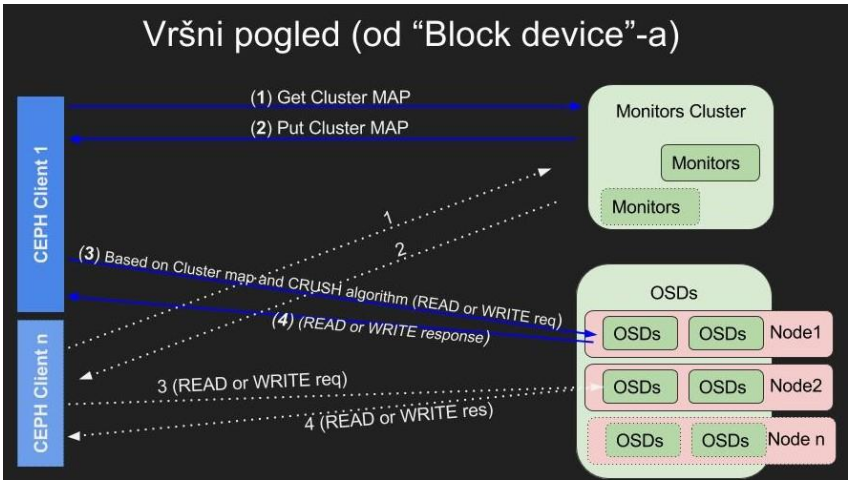
- **OpenStack** i
- **Proxmox VE**

Platforma za virtualizaciju za svako virtualno računalo koje koristi virtualni tvrdi disk (koji je zapravo *blok uređaj* tj. logički tvrdi disk od cijelog CEPH klastera), koristi **QEMU** (i Linux **KVM**).

QEMU i Linux KVM su zaduženi za sve potrebne funkcionalnosti da bi se virtualizacija uopće mogla koristiti. Dakle oni simuliraju sve virtualne komponente svakog pojedinog virtualnog računala (Matična ploča i njen BIOS, CPU, mrežna kartica i njen BIOS, disk kontroler i njem BIOS te pripadajući virtualni tvrdi disk, …)

Qemu kao Hipervizor ima nadalje metodu za korištenje svakog pojedinog virtualnog diska koji se zapravo nalazi unutar CEPH klastera (kao *Block device*). QEMU se tada spaja kao klijent na CEPH klaster i to na točno određeni **CEPH Pool** te njega koristi kao da je "polje diskova" na nekom SAN sustavu (jer govorimo o upotrebi CEPH-a kao *Block device*a tj. kao **RBD**)

A sada pogledajmo kako to izgleda sa strane *CEPH Block Device*a odnosno blok uređaja, kao krajnje komponente, koja na kraju stvarno pristupa CEPH klasteru za čitanje ili zapisivanje podataka. Ovdje zapravo QEMU kao CEPH klijent pristupa CEPH polju :



Klijent 1 piše ili čita na ili sa CEPH RBD

1. Kod procesa čitanja ili pisanja na *Block device* tj. CEPH RBD ,klijent koji žali nešto zapisati ili pročitati iz CEPH clustera koji koristi kao blok uređaj (logički kao tvrdi disk), prvo kontaktira CEPH klaster i to MONitor poslužitelje i od njih traži *Cluster Map* tablicu/konfiguraciju.
2. CEPH cluster MONitor poslužitelj(i) mu šalju traženu tablicu/konfiguraciju
3. Na osnovi tablice/konfiguracije koju je dobio, klijent pomoću CRUSH algoritma traži od OSD poslužitelja i OSD diskova podatke za čitanje ili traži pisanje. Do točnih OSD poslužitelja i točno određenih OSD diskova je pomoću CRUSH algoritma izračunao koji su te od njih i traži/šalje podatke
4. S OSD-ova dobiva odgovor na traženi zahtjev (čitanje ili pisanje)

Klijent 2 piše ili čita na ili sa CEPH RBD

Ponavlja se proces kao i za prvog klijenta

Izvori informacija

https://en.wikipedia.org/wiki/Network-attached_storage : Wikipedia - NAS sustavi

https://en.wikipedia.org/wiki/Storage_area_network : Wikipedia - SAN sustavi

https://en.wikipedia.org/wiki/RAID : Wikipedia - RAID

https://en.wikipedia.org/wiki/Nested_RAID_levels : Wikipedia - Ugniježđena RAID polja

https://www.opensource-osijek.org/dokuwiki/wiki:knjige:uvod_u_linux : Uvod u Linux i Linux napredno (uz suglasnost autora)

https://www.howtogeek.com/193669/whats-the-difference-between-gpt-and-mbr-when-partitioning-a-drive/ : MBR i GPT

https://en.wikipedia.org/wiki/Master_boot_record : MBR

https://en.wikipedia.org/wiki/GUID_Partition_Table : GPT

https://www.opensource-osijek.org/dokuwiki/wiki:knjige:ceph-storage : CEPH storage - visoko dostupni, redundanti, klasterski, skalabilan storage sustav (uz suglasnost autora)

Više informacija o knjizi

Više informacija o knjizi možete dobiti na stranici :

https://www.opensource-osijek.org/wordpress/kratka-prica-o-nas-i-san-sustavima-i-malo-vise/

QR kod - Link na stranicu: