

Elektronski fakultet, Niš

Smer: Računarstvo i informatika, Softversko inženjerstvo

Predmet: **Sistemi za upravljanje bazama podataka**

Tema:

Oracle High Availability rešenja

Student: Lena Petrović 1295

Mentor: Doc. dr Aleksandar Stanimirović

Niš, Jun 2022

Uvod	3
Važnost dostupnosti	4
Troškovi zastoja	4
Uzroci zastoja	5
Oracle rešenja za visoku dostupnost	5
Oracle Flashback	5
Oracle Flashback Query	6
Upit za Oracle Flashback verzije	7
Oracle Flashback transakcioni upit	9
Oracle Flashback Table	9
Oracle Flashback Drop	10
Oracle Flashback Database	10
Oracle Flashback tačke vraćanja	10
Oracle Data Guard	11
DataGuard arhitektura	12
Oracle Recovery Manager (RMAN)	13
Oracle LogMiner	16
Fast Start Fault Recovery	17
Oracle RAC - Real Application Clusters	18
Reference	21

Uvod

Baza podataka visoke dostupnosti (High Availability database) je sistem baze podataka dizajniran da radi neprekidno bez prekida u usluzi. Dakle, greške i kvarovi u bazi podataka moraju se rukovati automatskim prelaskom na redundantne čvorove kada dođe do problema.

Dostupnost je stepen do kojeg su aplikacija i usluga baze podataka dostupne. Dostupnost se meri percepcijom korisnika aplikacije. Korisnici doživljavaju frustraciju kada su njihovi podaci nedostupni ili računarski sistem ne radi kako se očekivalo. Greške u performansama zbog upotrebe veće od očekivane stvaraju isti problem kao i kvar kritičnih komponenti u arhitekturi. Ako korisnik ne može da pristupi aplikaciji ili usluzi baze podataka, kaže se da je nedostupna. Generalno, termin zastoja (downtime) se koristi i odnosi se na periode kada sistem nije dostupan.

Korisnicima koji žele da njihovi sistemi budu uvek spremni da im služe potrebna je visoka dostupnost. Sistem koji je veoma dostupan je dizajniran da obezbedi neprekidne računarske usluge tokom bitnih vremenskih perioda, tokom velikog broja sati u danu i većine dana u nedelji. Ovo merenje je poznato i kao 24*365.

Pouzdanost, mogućnost povrata, blagovremeno otkrivanje grešaka i kontinuirani rad su primarne karakteristike svakog visoko dostupnog rešenja.

- Pouzdanost. Pouzdan hardver je jedna komponenta rešenja visoke dostupnosti. Pouzdan softver (uključujući bazu podataka, veb servere i aplikacije) jednako je ključan za implementaciju visoko dostupnog rešenja.
- Mogućnost povrata. Iako postoji mnogo načina za oporavak od kvara, važno je utvrditi koji tipovi kvarova se mogu pojaviti u okruženju visoke dostupnosti i kako se brzo oporaviti od tih kvarova kako bi se ispunili poslovni zahtevi.
- Blagovremeno otkrivanje greške. Ako komponenta u arhitekturi otkáže, brzo otkrivanje je od suštinskog značaja za oporavak od neočekivanog kvara. Ukoliko postoji mogućnost brzog oporavka od prekida rada, ako je potrebno dodatnih 90 minuta da se otkrije problem, situacija nije idealna. Nadgledanje rada okruženja zahteva pouzdan softver za brz pregled i mogućnost da obavesti administratora baze podataka o problemu.
- Kontinualni rad. Obezbeđivanje kontinuiranog pristupa podacima je od suštinskog značaja kada je vrlo malo ili nimalo zastoja prihvatljivo za obavljanje aktivnosti održavanja. Aktivnosti, kao što je premeštanje tabele na drugu lokaciju u bazi podataka treba da budu transparentne za korisnika u arhitekturi visoke dostupnosti.

Detaljnije, arhitektura visoke dostupnosti treba da ima sledeće karakteristike:

- Tolerancija kvarova tako da se obrada nastavi sa minimalnim prekidima ili bez njih
- Transparentnost za promene sistema, podataka i aplikacija
- Obezbeđene ugrađene preventivne mere
- Obezbeđeno aktivno praćenje i brzo otkrivanje kvarova
- Obezbeđena brzu mogućnost povrata
- Automatizovane operacije otkrivanja i oporavka
- Zaštita podataka da bi se smanjio ili sprečio gubitak i oštećenje podataka

Važnost dostupnosti

Važnost visoke dostupnosti varira u zavisnosti od aplikacije. Baze podataka i internet omogućili su saradnju širom sveta i deljenje informacija proširivanjem dometa aplikacija baza podataka kroz organizacije i zajednice.

Ovde se naglašava važnost visoke dostupnosti u rešenjima za upravljanje podacima. I mala preduzeća i globalna preduzeća imaju korisnike širom sveta kojima je potreban pristup podacima 24 sata dnevno. Bez ovog pristupa podacima, operacije se mogu zaustaviti, a prihod je izgubljen. I klijenti i korisnici postaju sve zavisniji od ovih rešenja. Ako kritična (ili bilo koja) aplikacija postane nedostupna, posao može biti u opasnosti. Posao bi mogao da izgubi prihode, dobije kazne i dobije loš publicitet koji ima trajni efekat na kupce i na cenu akcija kompanije.

Troškovi zastoja

Potreba za isporukom sve većeg nivoa dostupnosti nastavlja da se ubrzava kako preduzeća reinženjering svojih rešenja kako bi stekla konkurentsku prednost. Najčešće se ova nova rešenja oslanjaju na trenutni pristup kritičnim poslovnim podacima.

Kada podaci nisu dostupni, operacija može prestati da funkcioniše. Zastoji mogu dovesti do gubitka produktivnosti, gubitka prihoda, oštećenih odnosa sa kupcima, lošeg publiciteta i tužbi.

Nije uvek lako odrediti direktnu cenu zastoja. Ljuti kupci, loši zaposleni i loš publicitet su skupi, ali se ne mere direktno u valuti. S druge strane, izgubljeni prihod i zakonske kazne mogu se lako kvantifikovati. Cena zastoja može brzo da poraste u industrijama koje zavise od svojih rešenja za pružanje usluga.

Uzroci zastoja

Jedan od izazova u dizajniranju rešenja visoke dostupnosti je ispitivanje i adresiranje svih mogućih uzroka zastoja.

Važno je uzeti u obzir uzroke i neplaniranih i planiranih zastoja prilikom dizajniranja IT infrastrukture otporne na greške i otpornosti. Planirani zastoji mogu biti jednako ometajući za operacije kao i neplanirani zastoji, posebno u globalnim preduzećima koja podržavaju korisnike u više vremenskih zona.

Oracle rešenja za visoku dostupnost

Računarska okruženja konfigurisana da obezbede skoro punu dostupnost poznata su kao sistemi visoke dostupnosti. Takvi sistemi obično imaju redundantni hardver i softver koji čini sistem dostupnim uprkos kvarovima. Dobro dizajnirani sistemi visoke dostupnosti izbegavaju pojedinačne tačke kvara.

Oracle ima niz proizvoda i funkcija koje obezbeđuju visoku dostupnost u slučajevima neplaniranih zastoja ili planiranih zastoja:

- Oracle Real Application Clusters
- Oracle Data Guard
- Oracle Streams
- Oracle Flashback Technology
- Automatic Storage Management
- Recovery Manager
- Flash Recovery Area
- Oracle Security Features
- Fast-Start Fault Recovery
- LogMiner

Oracle Flashback

Flashback tehnologija pruža skup funkcija za pregled i “premotavanje” podataka napred-nazad u vremenu. Funkcije flešbeka nude mogućnost da se ispituju prethodne verzije objekata šeme, da se ispituju istorijski podaci, da se izvrši analiza promena i izvrši samouslužna popravka da bi se oporavila od logičke oštećenja dok je baza podataka na mreži.

Flashback tehnologija obezbeđuje brzu analizu i popravku ljudskih grešaka. Flashback pruža detaljnu analizu i popravku lokalizovanih oštećenja (na primer, brisanje pogrešne porudžbine korisnika). Flashback tehnologija takođe omogućava ispravljanje širih oštećenja, ali to čini brzo kako bi se izbegli dugi zastoji. Flashback tehnologija je jedinstvena za Oracle bazu podataka i podržava oporavak na svim nivoima uključujući red, transakciju, tabelu, prostor tabele i bazu podataka.

Oracle Flashback Query

Oracle Flashback Query pruža mogućnost da se pregledaju podaci kakvi su postojali u prošlosti korišćenjem sistema za automatsko poništavanje upravljanja za dobijanje metapodataka i istorijskih podataka za transakcije. Podaci o opozivanju su trajni i preživeće kvar ili gašenje baze podataka. Jedinstvene karakteristike Flashback Query-a ne samo da pružaju mogućnost upita za prethodne verzije tabela, već takođe pružaju moćan mehanizam za oporavak od pogrešnih operacija.

Upotreba Flashback Query-ja podrazumeva:

- Oporavak izgubljenih podataka ili poništavanje netačnih, unesenih promena. Na primer, redovi koji su izbrisani ili ažurirani mogu se odmah popraviti čak i nakon što su upisani.
- Poređenje trenutnih podataka sa odgovarajućim podacima u nekom trenutku u prošlosti. Na primer, korišćenjem dnevnog izveštaja koji prikazuje promene u podacima od juče, moguće je uporediti pojedinačne redove podataka tabele, ili pronaći preseke ili spojeve skupova redova.
- Provera stanja transakcijskih podataka u određeno vreme, kao što je provera stanja na računu određenog dana.
- Pružanje samouslužne ispravke grešaka za aplikaciju, omogućavajući korisnicima da ponište i isprave svoje greške.

Sledi primer. Radi se nad tabelom Persons, konkretno nad objektima koji imaju personid veći od 5.

```
SQL> select count (*) from system.Persons where personid > 5;

COUNT(*)
-----
5
```

Recimo da se slučajno desi da je korisnik obrisao sve objekte ove tabele koji imaju personid veći od 5. Pretpostavlja se da je došlo do greške.

```
SQL> delete from system.Persons where personid > 5;

5 rows deleted.
```

```
SQL> select count (*) from system.Persons where personid > 5;

COUNT(*)
-----
0
```

Petnaest minuta nakon brisanja korisnik shvati da je napravio grešku i da su mu zapravo potrebni obrisani podaci. Sledećom select naredbom mogu se videti svi objekti tabele Persons

sa personid većim od 5, koji su se do pre 15 minuta nalazili u tabeli (*systimestamp - interval '15' minute*).

```
SQL> select * from system.Persons as of timestamp (systimestamp - interval '15' minute) where personid > 5;
```

Sada je potrebno napraviti 'kopiju' Persons tabele, ali onakve kakva je ona bila pre 15 minuta.

```
SQL> create table system.Persons_back as select * from system.Persons as of timestamp (systimestamp - interval '15' minute) where personid > 5;
Table created.
```

Vidi se da su pogrešno obrisani redovi povraćeni:

```
SQL> select count (*) from system.Persons_back where personid > 5;

COUNT(*)
-----
5
```

Zatim, baš ove izbrisane redove treba insert-ovati nazad u originalnu tabelu:

```
SQL> insert into system.Persons (select * from system.Persons as of timestamp (systimestamp - interval '15' minute) where personid > 5);
5 rows created.
```

Dokaz da su u originalnu tabelu vraćeni prethodno obrisani podaci:

```
SQL> select count (*) from system.Persons where personid > 5;

COUNT(*)
-----
5
```

Upit za Oracle Flashback verzije

Oracle Flashback Versions Query je proširenje za SQL koji se može koristiti za preuzimanje verzija redova u datoj tabeli koje su postojale u određenom vremenskom intervalu. Oracle Flashback Versions Query vraća red za svaku verziju reda koja je postojala u navedenom vremenskom intervalu. Za bilo koju datu tabelu, nova verzija reda se kreira svaki put kada se izvrši naredba COMMIT.

Upit za verzije Flashback-a je moćan alat za DBA da izvrši analizu kako bi utvrdio šta se dogodilo. Pored toga, programeri aplikacija mogu da koriste Flashback Versions Query za pravljenje prilagođenih aplikacija za potrebe revizije.

Kreiranje tabele nad kojom će biti realizovana demonstracija:

```
SQL> CREATE TABLE flashback_version_query_test (
  2   id          NUMBER(10),
  3   description VARCHAR2(50)
  4 );
Table created.
```

Dodavanje podataka:

```
SQL> INSERT INTO flashback_version_query_test (id, description) VALUES (1, 'ONE');
1 row created.
```

Provera SCN parametra i trenutnog vremena:

```
SQL> SELECT current_scn, TO_CHAR(SYSTIMESTAMP, 'YYYY-MM-DD HH24:MI:SS') FROM v$database;
CURRENT_SCN TO_CHAR(SYSTIMESTAM
-----
2877657 2022-06-11 19:41:21
```

Ažuriranje jednog jedinog podatka više puta:

```
SQL> UPDATE flashback_version_query_test SET description = 'TWO' WHERE id = 1;
1 row updated.
```

```
SQL> UPDATE flashback_version_query_test SET description = 'THREE' WHERE id = 1;
1 row updated.
```

Ponovna provera SCN parametra i trenutnog vremena:

```
SQL> SELECT current_scn, TO_CHAR(SYSTIMESTAMP, 'YYYY-MM-DD HH24:MI:SS') FROM v$database;
CURRENT_SCN TO_CHAR(SYSTIMESTAM
-----
2877672 2022-06-11 19:41:52
```

Prikaz verzija podataka u toku vremena između dva SCN trenutka:

```
SQL> SELECT versions_startscn, versions_starttime,
2     versions_endscn, versions_endtime,
3     versions_xid, versions_operation,
4     description
5 FROM   flashback_version_query_test
6     VERSIONS BETWEEN SCN 2877657 AND 2877672
7 WHERE  id = 1;
```

VERSIONS_STARTSCN	VERSIONS_STARTTIME	VERSIONS_ENDSCN	VERSIONS_ENDTIME	VERSIONS_XID	V DESCRIPTION
2877667	11-JUN-22 07.41.41 PM	2877667	11-JUN-22 07.41.41 PM	06000B002B050000	U THREE
2877663	11-JUN-22 07.41.35 PM	2877663	11-JUN-22 07.41.35 PM	0A0012002B050000	U TWO
					ONE

Oracle Flashback transakcioni upit

Oracle Flashback Transaction Query pruža mehanizam za pregled svih promena napravljenih u bazi podataka na nivou transakcije. Kada se koristi zajedno sa Flashback Versions Query-jem, nudi brzo i efikasno sredstvo za oporavak od greške korisnika ili aplikacije. Flashback Transaction Query povećava mogućnost obavljanja onlajn dijagnostike problema u bazi podataka vraćanjem korisnika koji je promenio red i obavljanjem analize i revizije transakcija.

Oracle Flashback Table

Oracle Flashback Table omogućava korisnicima da povrate tabelu na prethodni vremenski trenutak. Pruža brzo, onlajn rešenje za oporavak tabele ili skupa tabela koje je korisnik ili aplikacija greškom izmenio. U većini slučajeva, Flashback Table ublažava potrebu da administratori obavljaju komplikovanije operacije oporavka u trenutku. Čak i nakon flešbeka, podaci u originalnoj tabeli se ne gube.

Važna stvar u vezi sa flešbek operacijom je da poništavanje podataka kontroliše parametar UNDO_RETENTION. Jednostavnije rečeno, ako su dostupni undo podaci, onda se samo tabele mogu vratiti nazad.

Jednom kada se pokrene naredba FLASHBACK TABLE na bilo kojoj tabeli, ona se ne može vratiti nazad.

Mora se omogućiti pomeranje redova tokom kreiranja tabele da bi se izvršila flešbek operacija nad njom. Pomeranje reda se takođe može omogućiti korišćenjem komande ALTER (*alter table TableName enable row movement*).

Brisanje određenih redova iz tabele:

```
SQL> delete from system.Persons where personid > 5;
10 rows deleted.
```

Vraćanje tabele Persons na 2 minuta pre trenutnog vremena korišćenjem flashback operacije:

```
SQL> flashback table system.Persons to timestamp (systimestamp - interval '2' minute);
Flashback complete.
```

Vraćanje tabele Persons na specificirani vremenski trenutak pre trenutnog vremena korišćenjem flashback operacije:

```
SQL> flashback table system.Persons to timestamp to_timestamp('2022-06-11 20:00:00', 'YYYY-MM-DD HH24:MI:SS');
Flashback complete.
```

Takođe se može koristiti SCN broj za ovo vraćanje. Treba ga samo zapamtiti u određenom trenutku u prošlosti.

Oracle Flashback Drop

Slučajno brisanje objekata je uvek predstavljalo problem i za korisnike i za DBA. Istorijski gledano, ne postoji jednostavan način da se povrate obrisane tabele, indeksi, ograničenja ili trigeri. Oracle Flashback Drop pruža zaštitnu mrežu prilikom brisanja predmeta. Kada korisnik

obriše tabelu, Oracle je automatski stavlja u korpu za otpatke. Korpa za otpatke je virtuelni kontejner u kome se nalaze svi obrisani predmeti. Korisnici mogu da nastavu da traže podatke u tabeli (dropped table).

```
SQL> drop table recipes;  
  
Table dropped.  
  
SQL> flashback table recipes to before drop;
```

Oracle Flashback Database

Oracle Flashback Database pruža efikasniju alternativu oporavku baze podataka u trenutku. Sa Oracle Flashback Database, trenutne datoteke podataka se mogu vratiti na njihov sadržaj u prošlom trenutku. Rezultat je sličan rezultatu oporavka u trenutku kada se koriste rezervne kopije datoteka sa podacima i redo evidencije, ali nije neophodno vraćati datoteke podataka iz rezervne kopije ili ponovo primeniti onoliko pojedinačnih promena u redo evidencijama koliko je potrebno u konvencionalnim oporavak medija.

Omogućavanje Oracle Flashback baze podataka pruža sledeće prednosti:

- Eliminacija vremena za vraćanje rezervne kopije kada se ispravljaju ljudske greške.
- Omogućava bazama podataka u stanju pripravnosti da koriste primenu u realnom vremenu za sinhronizaciju sa primarnom bazom podataka jer ljudske greške mogu brzo da se ponište.
- Omogućava brzo ponovno pokretanje baze podataka u stanju pripravnosti nakon prelaska na grešku u bazi podataka.

Oracle Flashback tačke vraćanja

Kada se operacija Oracle Flashback oporavka izvrši na bazi podataka, DBA mora odrediti tačku u vremenu na koju se podaci mogu kasnije vratiti nazad. Oracle Flashback tačke vraćanja su korisnički definisane oznake koje se mogu zameniti za SCN ili vreme transakcije koje se koristi u operacijama Flashback Database, Flashback Table i Recovery Manager (RMAN). Štaviše, baza podataka se može vratiti nazad kroz prethodni oporavak baze podataka i otvoriti evidenciju resetovanja korišćenjem garantovanih tačaka vraćanja. Garantovane tačke vraćanja omogućavaju da se velike promene baze podataka — kao što su grupni poslovi baze podataka, nadogradnja ili zakrpa — brzo ponište tako što će obezbediti da se poništavanje potrebno za premotavanje baze podataka zadrži.

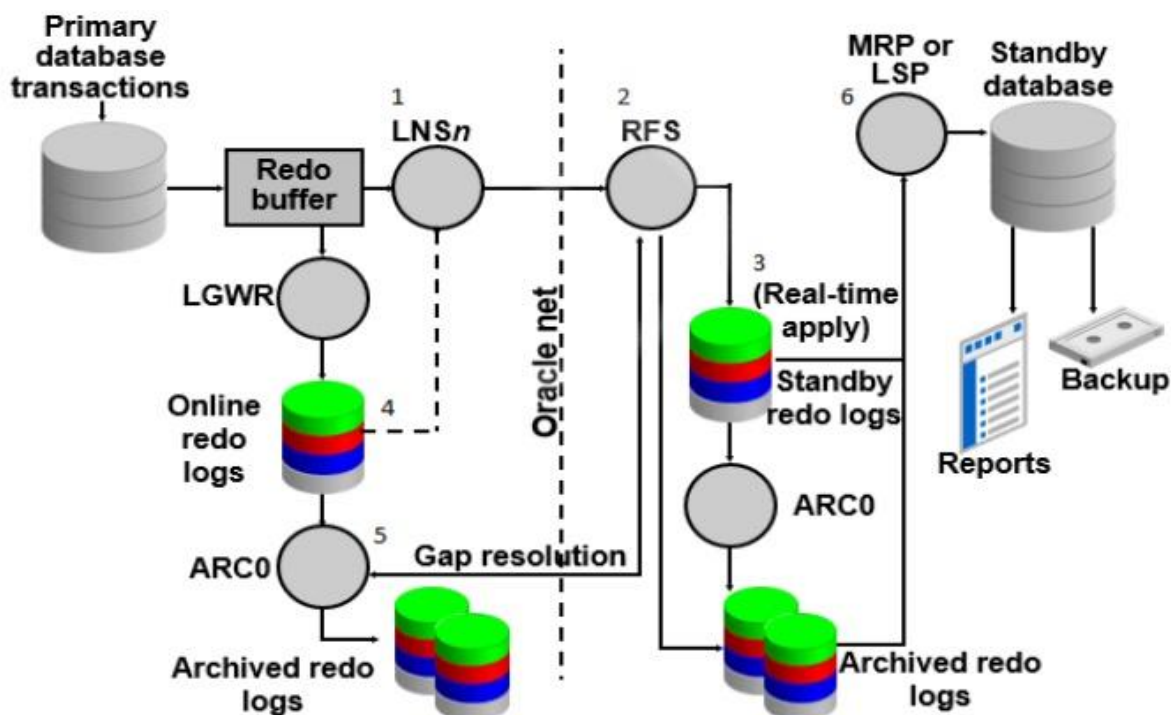
Oracle Data Guard

Oracle Data Guard obezbeđuje visoku dostupnost, zaštitu podataka i oporavak od katastrofe za podatke. Data Guard pruža sveobuhvatan skup usluga koje kreiraju, održavaju, upravljaju i nadgledaju jednu ili više baza podataka u stanju pripravnosti kako bi se glavnim (proizvodnim) Oracle bazama podataka omogućilo da prežive katastrofe i oštećenje podataka. Data Guard održava ove rezervne baze podataka kao transakciono konzistentne kopije glavne baze podataka. Zatim, ako glavna baza podataka postane nedostupna zbog planiranog ili neplaniranog prekida rada, Data Guard može prebaciti bilo koju bazu podataka u stanju pripravnosti na glavnu ulogu, minimizirajući vreme zastoja povezano sa prekidom rada. Data Guard se može koristiti sa tradicionalnim tehnikama pravljenja rezervnih kopija, obnavljanja i klastera kako bi se obezbedio visok nivo zaštite podataka i dostupnosti podataka.

Oracle Data Guard pruža sledeće prednosti:

- Održava transakciono konzistentne kopije baze podataka u realnom vremenu radi zaštite od neplaniranih zastoja i katastrofe
- Potpuna zaštita podataka od kvarova računara, ljudskih grešaka, oštećenja podataka i grešaka na sajtu
- Smanjuje planirano vreme zastoja za nadogradnju hardvera i sistema
- Automatsko otkrivanje i rešavanje podataka koji nedostaju nakon privremenog gubitka veze između primarne i rezervne baze podataka
- Više nivoa zaštite podataka i performansi za balansiranje dostupnosti podataka sa zahtevima performansi sistema
- Omogućava efikasno korišćenje sistemskih resursa preusmeravanjem izveštaja i operacija rezervnih kopija iz glavne baze podataka u baze podataka u stanju pripravnosti
- Upravljana i automatska tranzicija uloga i obaveštenja o aplikaciji kako bi se minimizirali planirani i neplanirani zastoji
- Automatska resinhronizacija neuspele primarne baze podataka nakon prelaska na grešku
- Svi sistemi upravljani kao jedna konfiguracija za pojednostavljenu administraciju

DataGuard arhitektura



Na slici iznad je prikazana arhitektura Oracle Data Guard-a. Treba razlikovati levu i desnu stranu, sa leve se nalazi primarna baza podataka, dok se sa desne nalazi baza podataka koja je u pripravnosti. Ukoliko bude bilo potrebe, izvršenje će se sa primarne baze prebaciti na bazu podataka u pripravnosti.

Sa strane primarnog servera, pre svega se nalazi primarna baza podataka. Nakon toga moguće je videti Redo log bafer, koji skladišti promene napravljene nad bazom podataka. Redo log se generiše od strane svake transakcije izvršene nad bazom podataka.

Zatim postoji proces LGWR (Log Writer Process) koji prenosi podatke iz Redo Log bafera u Online redo logs fajlove.

Sada iz Online redo log fajlova, nastaje proces arhiviranja koji je zadužen za upisivanje log fajlova u Archived redo log fajlove.

Ovo je proces koji se dešava sa strane primarne baze podataka i osnovno je ponašanje kod svake samostalne baze podataka.

Kada se radi o Data Guard-u postoji još jedan proces koji direktno čita podatke iz Redo log bafera. Ovaj proces je poznat kao Log Network Service - LNS i odgovoran je za čitanje podataka iz Redo Log bafera primarne baze i slanje preko mreže do baze podataka u pripravnosti. Oracle Net Service je naziv mreže preko koje se vrši slanje.

Sa strane baze podataka u pripravnosti mora postojati neki proces koji će prihvatiti podatke poslate preko mreže. To je Remote Fetch Service proces - RFS. Kombinacija LNS i RFS je poznata kao Long Shipping Service - LSS.

Postavlja se pitanje šta radi RFS kada do njega stignu fajlovi primarne baze? Njegovo zaduženje je upis Redo Log fajlova u "Standby Redo Log" fajlove koji su na strani baze podataka u pripravnosti.

Da bi se sve ovo povezalo mora postojati nešto što će primeniti podatke pročitane iz "Standby Redo Log" na bazu podataka u pripravnosti. Proces čije je ovo zaduženje naziva se MRP - Managed Recovery Process.

Međutim, može doći do nepredviđenih situacije. Recimo da LNS proces nije dovoljno brz i da ne stiže da prosledi sve informacije o Redo Log-ovima na Oracle Net. Ukoliko dođe do ovoga LNS proces će podatke čitati iz Online redo log fajlova i slati na mrežu. Ili ukoliko podaci odu u Archived redo log files pre nego što su poslali bazi podataka u pripravnosti, tada RFS proces direktno kontaktira proces koji arhivira podatke. Nakon ovoga podaci se šalju iz Archive redo logs primarne strane na stranu koja je u pripravnosti. Ovaj proces je poznat kao "Gap resolution".

Kao što je rečeno, Oracle baza podataka radi u jednoj od dve uloge: primarnoj ili pripravnoj. Data Guard pomaže kod promene uloge baze podataka pomoću prebacivanja ili preko greške (switchover ili failover):

- Switchover - zamena uloga između primarne baze podataka i jedne od njenih rezervnih baza podataka. Tokom prelaska, primarna baza podataka prelazi u ulogu pripravnosti, a rezervna baza podataka prelazi u primarnu ulogu.
- Failover - vrši se kada primarna se baza podataka pokvari ili postane nedostupna i jedna od rezervnih baza podataka se prebaci da preuzme primarnu ulogu.

Oracle Recovery Manager (RMAN)

RMAN je samostalni uslužni program za upravljanje backup-om i oporavkom. Predstavlja veoma moćan alat koji dolazi uz instalaciju Oracle baze podataka. Podrazumevano, RMAN kreira rezervne kopije na disku i generiše skupove rezervnih kopija umesto kopija slika.

Slede koraci za kreiranje rezervne kopije. Pre svega ORACLE_SID mora biti podešen na bazu podataka koju treba backup-ovati. Ta baza podataka mora biti postavljena (mount-ovana) i biti u ARCHIVELOG modu.

```
C:\Users\Korisnik\Downloads\WINDOWS.X64_193000_db_home\bin>set ORACLE_SID=ORCL
C:\Users\Korisnik\Downloads\WINDOWS.X64_193000_db_home\bin>echo %ORACLE_SID%
ORCL
```

Kada su svi preduslovi ispunjeni treba se povezati na RMAN (slično kao povezivanje na SQLPlus), komandom na slici ispod:

```
C:\Users\Korisnik\Downloads\WINDOWS.X64_193000_db_home\bin>rman target /  
Recovery Manager: Release 19.0.0.0.0 - Production on Wed Jun 15 13:06:24 2022  
Version 19.3.0.0.0  
  
Copyright (c) 1982, 2019, Oracle and/or its affiliates. All rights reserved.  
connected to target database: ORCL (DBID=1629254697)
```

Kreiranje rezervne kopije je veoma jednostavno. Treba pokrenuti komandu “backup database” i kreiranje će početi.

Oracle RMAN skladišti podatke u datotekama (image files) ili skupovima rezervnih setova (backup set), koji se sastoje od rezervnih delova (backup pieces). Backup piece je binarna datoteka specifična za RMAN koju samo RMAN može da kreira ili vrati. Backup pieces su grupisani u backup set-ove, omogućavajući DBA-ima da zajedno štite više datoteka sa podacima, kontrolnih datoteka, fajlova parametara servera i arhivskih dnevnika. RMAN može da šifrjuje i dešifrjuje podatke upisane u rezervne skupove.

U prvom delu se može videti koji su to fajlovi čija se rezervna kopija kreira (to su fajlovi sa ekstenzijom .dbf).

Kreirana su tri backup set-a. Backup piece je ime fajla u koji je backup set upisan. Fajlovi koji sadrže rezervne podatke su 011048T5_1_1, 021048US_1_1 i 031048VL_1_1. Naziv se sastoji od 8 automatski generisanih karaktera i karaktera _1_1 na kraju. Svaki fajl će imati sekvencu _1_1 ukoliko se radi preko RMAN-a.




```

RMAN> backup database;

Starting backup at 15-JUN-22
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=143 device type=DISK
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00001 name=C:\USERS\KORISNIK\DOWNLOADS\ORADATA\ORCL\SYSTEM01.DBF
input datafile file number=00003 name=C:\USERS\KORISNIK\DOWNLOADS\ORADATA\ORCL\SYS_AUX01.DBF
input datafile file number=00004 name=C:\USERS\KORISNIK\DOWNLOADS\ORADATA\ORCL\UNDOTBS01.DBF
input datafile file number=00013 name=C:\USERS\KORISNIK\DOWNLOADS\WINDOWS.X64_193000_DB_HOME\DATABASE\NORM_DATA.DBF
input datafile file number=00014 name=C:\USERS\KORISNIK\DOWNLOADS\WINDOWS.X64_193000_DB_HOME\DATABASE\TDE_DATA.DBF
input datafile file number=00015 name=C:\USERS\KORISNIK\DOWNLOADS\WINDOWS.X64_193000_DB_HOME\DATABASE\FUTURE_DATA.DBF
input datafile file number=00007 name=C:\USERS\KORISNIK\DOWNLOADS\ORADATA\ORCL\USERS01.DBF
channel ORA_DISK_1: starting piece 1 at 15-JUN-22
channel ORA_DISK_1: finished piece 1 at 15-JUN-22
piece handle=C:\USERS\KORISNIK\DOWNLOADS\WINDOWS.X64_193000_DB_HOME\DATABASE\011048T5_1_1 tag=TAG20220615T131412 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:55
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00010 name=C:\USERS\KORISNIK\DOWNLOADS\ORADATA\ORCL\ORCLPDB\SYS_AUX01.DBF
input datafile file number=00009 name=C:\USERS\KORISNIK\DOWNLOADS\ORADATA\ORCL\ORCLPDB\SYSTEM01.DBF
input datafile file number=00011 name=C:\USERS\KORISNIK\DOWNLOADS\ORADATA\ORCL\ORCLPDB\UNDOTBS01.DBF
input datafile file number=00012 name=C:\USERS\KORISNIK\DOWNLOADS\ORADATA\ORCL\ORCLPDB\USERS01.DBF
channel ORA_DISK_1: starting piece 1 at 15-JUN-22
channel ORA_DISK_1: finished piece 1 at 15-JUN-22
piece handle=C:\USERS\KORISNIK\DOWNLOADS\WINDOWS.X64_193000_DB_HOME\DATABASE\021048US_1_1 tag=TAG20220615T131412 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:25
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00006 name=C:\USERS\KORISNIK\DOWNLOADS\ORADATA\ORCL\PDBSEED\SYS_AUX01.DBF
input datafile file number=00005 name=C:\USERS\KORISNIK\DOWNLOADS\ORADATA\ORCL\PDBSEED\SYSTEM01.DBF
input datafile file number=00008 name=C:\USERS\KORISNIK\DOWNLOADS\ORADATA\ORCL\PDBSEED\UNDOTBS01.DBF
channel ORA_DISK_1: starting piece 1 at 15-JUN-22
channel ORA_DISK_1: finished piece 1 at 15-JUN-22
piece handle=C:\USERS\KORISNIK\DOWNLOADS\WINDOWS.X64_193000_DB_HOME\DATABASE\031048VL_1_1 tag=TAG20220615T131412 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:15
Finished backup at 15-JUN-22

Starting Control File and SPFILE Autobackup at 15-JUN-22
piece handle=C:\USERS\KORISNIK\DOWNLOADS\WINDOWS.X64_193000_DB_HOME\DATABASE\C-1629254697-20220615-00 comment=NONE

```

	031048VL_1_1	6/15/2022 1:15 PM	File	477,176 KB
	021048US_1_1	6/15/2022 1:15 PM	File	461,032 KB
	011048T5_1_1	6/15/2022 1:15 PM	File	1,338,424 KB

Za pregled svih rezervnih kopija koristi se komanda “list backup”. Izvršenje je prikazano u nastavku. Prikazan je samo piece 011048T5_1_1 i detalji vezani za njega.

```

RMAN> list backup;

List of Backup Sets
=====

BS Key   Type LV Size       Device Type Elapsed Time Completion Time
-----
1        Full  1.28G   DISK        00:00:50    15-JUN-22
BP Key: 1  Status: AVAILABLE Compressed: NO Tag: TAG20220615T131412
Piece Name: C:\USERS\KORISNIK\DOWNLOADS\WINDOWS.X64_193000_DB_HOME\DATABASE\011048T5_1_1

List of Datafiles in backup set 1
File LV Type Ckp SCN      Ckp Time    Abs Fuz SCN Sparse Name
-----
1        Full  2900249   15-JUN-22   NO          C:\USERS\KORISNIK\DOWNLOADS\ORADATA\ORCL\SYSTEM01.DBF
3        Full  2900249   15-JUN-22   NO          C:\USERS\KORISNIK\DOWNLOADS\ORADATA\ORCL\SYS_AUX01.DBF
4        Full  2900249   15-JUN-22   NO          C:\USERS\KORISNIK\DOWNLOADS\ORADATA\ORCL\UNDOTBS01.DBF
7        Full  2900249   15-JUN-22   NO          C:\USERS\KORISNIK\DOWNLOADS\ORADATA\ORCL\USERS01.DBF
13       Full  2900249   15-JUN-22   NO          C:\USERS\KORISNIK\DOWNLOADS\WINDOWS.X64_193000_DB_HOME\DATABASE\NORM_DATA.DBF
14       Full  2900249   15-JUN-22   NO          C:\USERS\KORISNIK\DOWNLOADS\WINDOWS.X64_193000_DB_HOME\DATABASE\TDE_DATA.DBF
15       Full  2900249   15-JUN-22   NO          C:\USERS\KORISNIK\DOWNLOADS\WINDOWS.X64_193000_DB_HOME\DATABASE\FUTURE_DATA.DBF

```

Brisanje rezervnih kopija:

```

RMAN> delete backup;

using channel ORA_DISK_1

List of Backup Pieces
BP Key   BS Key   Pc#   Cp#   Status   Device Type Piece Name
-----
1        1        1     1     AVAILABLE DISK        C:\USERS\KORISNIK\DOWNLOADED\WINDOWS.X64_193000_DB_HOME\DATABASE\011048T5_1_1
2        2        1     1     AVAILABLE DISK        C:\USERS\KORISNIK\DOWNLOADED\WINDOWS.X64_193000_DB_HOME\DATABASE\021048US_1_1
3        3        1     1     AVAILABLE DISK        C:\USERS\KORISNIK\DOWNLOADED\WINDOWS.X64_193000_DB_HOME\DATABASE\031048VL_1_1
4        4        1     1     AVAILABLE DISK        C:\USERS\KORISNIK\DOWNLOADED\WINDOWS.X64_193000_DB_HOME\DATABASE\C-1629254697-20220615-00

Do you really want to delete the above objects (enter YES or NO)?

```

Restore baze podataka:

```

RMAN> restore database;

Starting restore at 15-JUN-22
using target database control file instead of recovery catalog

```

Recover baze podataka:

```

RMAN> recover database;

Starting recover at 15-JUN-22
using channel ORA_DISK_1

starting media recovery

```

Oracle LogMiner

Oracle datoteke evidencije (log fajlovi) sadrže korisne informacije o aktivnostima i istoriji Oracle baze podataka. Log fajlovi sadrže sve podatke neophodne za oporavak baze podataka, a takođe beleže sve promene napravljene u podacima i metapodacima unutar baze podataka.

LogMiner je alat koji omogućava korišćenje SQL naredbi za analizu događaja u log fajlu baze podataka. Pomoću LogMiner-a se mogu pratiti transakcije ili locirati određene funkcije koje dovode do modifikacija podataka. LogMiner se može koristiti, zajedno sa podacima revizije, da se utvrdi šta se dogodilo u Oracle bazi podataka.


```
SQL> SELECT scn,
           username,
           sql_redo,
           sql_undo
        FROM V$LOGMNR_CONTENTS;
```

SCN	USERNAME	SQL_REDO	SQL_UNDO
232009	LIAR_LIAR	set transaction read write;	
232009	LIAR_LIAR	delete from SCOTT.EMP where	insert into
		EMPNO = 7902 and ENAME =SCOTT	EMP(EMPNO,ENAME,JOB,MGR,
		'ford' and JOB = 'ANALYST' and HIREDATE,SAL,COMM,DEPTNO)	
		MGR = 7566 and HIREDATE =	values
		TO_DATE('03-DEC-2081	(7902,'ford','ANALYST',
		00:00:00', 'DD-MON-YYYY	7566,TO_DATE('03-DEC-2081
		HH24:MI:SS') and SAL = 3000	00:00:00','DD-MON-YYYY
		and COMM IS NULL and DEPTNO =	HH24:MI:SS'),3000,NULL,
		20 and ROWID =	20);
		'AAAAAxAADAAAF6AAM';	
232010	LIAR_LIAR	commit;	

Fast Start Fault Recovery

Oracle baza podataka pruža brz i predvidljiv oporavak od sistemskih grešaka i otkaza baze podataka. Tehnologija Fast-Start Fault Recovery uključena u Oracle Database automatski ograničava vreme oporavka instance pri pokretanju korišćenjem samopodešene obrade kontrolnih tačaka. Ovo čini vreme oporavka brzim i predvidljivim. Funkcija Oracle Fast-Start Fault Recoveri može smanjiti vreme oporavka na jako opterećenoj bazi podataka sa desetina minuta na nekoliko sekundi.

Funkcije brzog pokretanja oporavka od greške uključuju:

- Predvidljiv, ograničen oporavak od grešaka instance, baze podataka i računara
- Kontrolna tačka baze podataka koja se samopodešava radi održavanja željenog cilja vremena oporavka

Oracle nudi funkciju brzog pokretanja oporavka od greške za kontrolu oporavka instance. Ovo smanjuje vreme potrebno za oporavak keša i čini oporavak ograničenim i predvidljivim ograničavanjem broja prljavih bafera i broja ponavljanih zapisa generisanih između poslednjeg zapisa ponavljanja i poslednje kontrolne tačke.

Osnova brzog oporavka je arhitektura kontrolnih tačaka sa brzim startom. Kontrolna tačka brzog pokretanja se javlja postepeno. Postoje DBW procesi koji periodično upisuju bafere na disk kako bi se unapredila pozicija kontrolne tačke. Najstariji modifikovani blokovi se prvo pišu kako bi se osiguralo da svako upisivanje omogućava napredovanje kontrolne tačke.

Administratori određuju ograničeno vreme za završetak faze oporavka iz keš memorije pomoću parametra FAST_START_MTTR_TARGET, a Oracle automatski menja inkrementalnu kontrolnu tačku upisivanja kako bi ispunio taj cilj. Parametar inicijalizacije FAST_START_MTTR_TARGET omogućava da se u sekundama navede očekivano „srednje vreme za oporavak“, što je očekivana količina vremena potrebnog Oracle-u da izvrši pad ili oporavak instance za jednu instancu.

Oracle RAC - Real Application Clusters

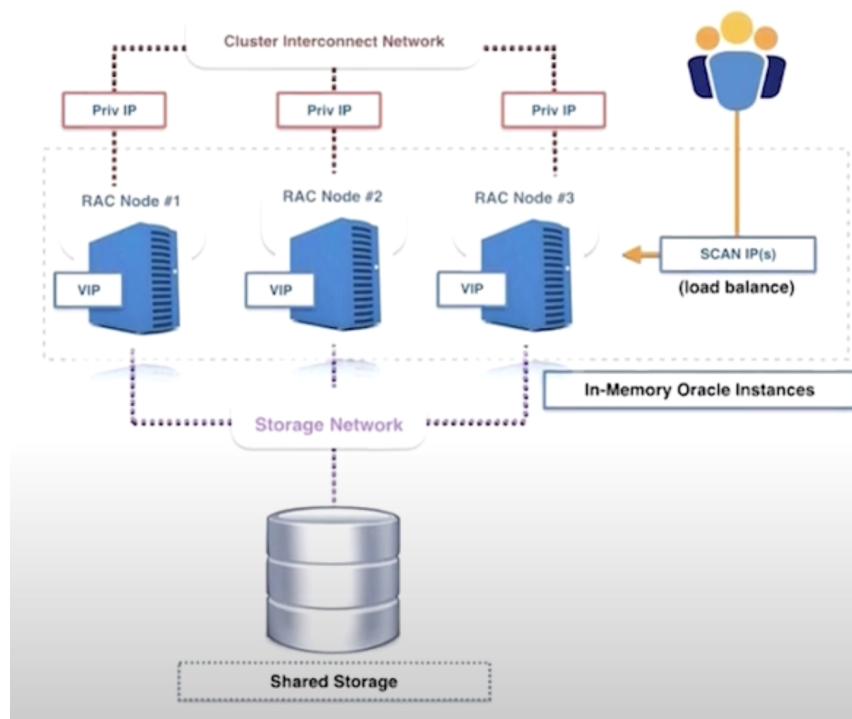
Klaster se sastoji od više međusobno povezanih računara ili servera koji krajnjim korisnicima i aplikacijama izgledaju kao da su jedan server. Oracle RAC omogućava da se grupiše Oracle baza podataka. Oracle RAC koristi Oracle Clusterware za infrastrukturu za povezivanje više servera kako bi oni funkcionisali kao jedan sistem.

Oracle Clusterware je prenosivo rešenje za upravljanje klasterima koje je integrisano sa Oracle bazom podataka. Oracle Clusterware je takođe neophodna komponenta za korišćenje Oracle RAC-a. Pored toga, Oracle Clusterware omogućava i neklasterskim Oracle bazama podataka i Oracle RAC bazama podataka da koriste Oracle infrastrukturu visoke dostupnosti. Oracle Clusterware vam omogućava da kreirate klasterizovani skup skladišta koji će koristiti bilo koja kombinacija neklasternih i Oracle RAC baza podataka.

Oracle RAC baza podataka se po arhitekturi razlikuje od neklasterske Oracle baze podataka po tome što svaka instanca Oracle RAC baze podataka takođe ima:

- Najmanje jedna dodatna nit ponavljanja za svaku instancu
- Tabela prostor za poništavanje specifičan za instancu

Kombinovana procesorska snaga više servera može da obezbedi veću propusnost i Oracle RAC skalabilnost nego što je to dostupno sa jednog servera.



Slika iznad pokazuje kako Oracle RAC obezbeđuje jednu sliku sistema za više servera za pristup jednoj Oracle bazi podataka. U Oracle RAC-u, svaka Oracle instanca mora da radi na zasebnom serveru.

Oracle RAC omogućava višestrukim instancama da pristupe jednoj Oracle bazi podataka. Ove instance često rade na više čvorova (koji su na slici obeleženi sa RAC Node #1, RAC Node #2 i RAC Node #3). Veći broj ovih čvorova deli jedno skladište.

Ovde se radi o tome da se razdvoji celokupno opterećenje na više različitih čvorova. Čvorovi će komunicirati jedni sa drugima putem privatne interkonekcije (Priv IP).

Korisnici mogu da komuniciraju preko baze podataka koristeći SCAN IP tako da će balansi opterećenja obavljati ovaj zadatak. Scan ustvari predstavlja string konekcije (connection string), tako da se korisnik ne povezuje direktno sa instancom. Korisnik je potpuno nesvestan čvora na kome se zahtev ispunjava i koji mu je dodeljen.

Dva bitna svojstva Oracle RAC arhitekture su visoka dostupnost i skalabilnost. Što se tiče visoke dostupnosti, korisnici mogu pristupiti bazi podataka bez čekanja da baza podataka bude dostupna na određenom čvoru. Bitno je da se ne oslanja direktno na konkretnu instancu baze podataka da postane aktivna. Kada je skalabilnost u pitanju, broj korisnika se može povećati dodavanjem dodatnih čvorova.

U samoj RAC arhitekturi, komponente se dele na komponente koje su dele (su zajedničke) i one koje to nisu. Neke deljene komponente su data fajlovi, konfiguracioni fajl, flash recovery log, postoje i deljena skladišta kao što su NAS i SAN.

Zatim postoje komponente kao što su pozadinski procesi (svaka instanca ima svoj skup pozadinskih procesa), arhivirana redo datoteka (privatno za instancu, ali drugim instancama će biti potreban pristup svim potrebnim arhivskim evidencijama ako sistem ili druga RAC administracija zahteva to), alert log i datoteke praćenja (ove datoteke su privatne za svaku instancu, druge instance nikada ne čitaju ili pišu u te datoteke).

Glavne komponente Oracle rac sistema:

- Sistem deljenih diskova

U deljenom skladištu, datoteke baze podataka treba da budu podjednako dostupne svim čvorovima istovremeno. Generički sistemi datoteka ne dozvoljavaju da se diskovi montiraju u više od jednog sistema.

Oracle RAC koristi arhitekturu „sve deljeno“, što znači da sve datoteke sa podacima, kontrolne datoteke, datoteke dnevnika ponavljanja i SPFILE moraju biti uskladištene na deljenim diskovima koji su svesni klastera. Čuvanje ovih datoteka u deljenoj memoriji osigurava da sve instance baze podataka koje su deo klastera mogu da pristupe različitim datotekama neophodnim za rad instance.

U prethodnim verzijama, Oracle je dozvoljavao da se koriste uređaji kao jedan od načina da se obezbedi deljeno skladište za Oracle RAC. Međutim, u Oracle Database 12c, uređaji nisu

sertifikovani za upotrebu u RAC okruženju. Umesto toga, mora se koristiti jedno od sledećih rešenja za skladištenje klastera koje podržava Oracle:

- Automatsko upravljanje skladištem (ASM) - Oracle preporučuje
- Sistem datoteka klastera
- Oracle clusterware

Oracle Clusterware softver obezbeđuje osnovnu infrastrukturu koja omogućava funkcionisanje Oracle RAC-a. Oracle Clusterware je taj koji upravlja entitetima kao što su baze podataka, slušaoci, usluge i virtuelne IP adrese, koje se nazivaju resursi. Oracle Clusterware koristi procese ili servise koji se pokreću na pojedinačnim čvorovima klastera da bi obezbedio mogućnosti visoke dostupnosti za RAC bazu podataka. Omogućava čvorovima da međusobno komuniciraju i formira klaster koji čini da čvorovi rade kao jedan logički server.

Oracle Clusterware se sastoji od nekoliko pozadinskih procesa koji obavljaju različite funkcije kako bi olakšali rad klastera. Oracle Clusterware je obavezan deo softvera potreban za pokretanje Oracle RAC opcije. Pruža osnovnu podršku za klasterisanje na nivou operativnog sistema i omogućava Oracle softveru da radi u režimu grupisanja.

- Cluster high-speed interconnect

Cluster interconnect je komunikaciono sredstvo sa visokim propusnim opsegom i malim kašnjenjem koje povezuje svaki čvor sa drugim čvorovima u klasteru i usmerava poruke između čvorova. Generalno, interkonekcija klastera se koristi za sledeće funkcije visokog nivoa:

- Nadgledanje zdravlja, statusa i sinhronizacija poruka
- Prenos poruka
- Pristup udaljenim sistemima datoteka
- Pomeranje saobraćaja specifičnog za aplikaciju

Interkonekcije takođe moraju biti sposobne da otkriju i izoluju kvarove i da koriste alternativne puteve.

Prednosti korišćenja RAC-a:

- Visoka dostupnost: Preživeti kvarove čvorova i instanci
- Skalabilnost: Dodavanje više čvorova ako budu potrebni u budućnosti
- Plaćajte kako rastete: plaćate samo ono što vam je potrebno danas

Reference

<https://docs.oracle.com/en/database/oracle/oracle-database/19/haovw/overview-of-ha.html#GUID-E142FDDA-80DF-4A49-A115-E1D50ADE88DE>
<https://www.youtube.com/watch?v=EFvQzylMnM>
https://docs.oracle.com/cd/B19306_01/server.102/b14210/hafeatures.htm#CJACDFJH
https://docs.oracle.com/cd/B19306_01/server.102/b14210/hafeatures.htm
<https://www.mydbaworld.com/oracle-flashback-query-restore-database-tspitr/>
<https://oracle-base.com/articles/10g/flashback-version-query-10g>
<https://dzone.com/articles/flashback-table-in-oracle-explained-with-examples>
<https://oracle-base.com/articles/10g/flashback-database-10g>
<https://oracle-base.com/articles/11g/data-guard-setup-11gr2>
https://www.youtube.com/watch?v=KXsXje_ffZ0&ab_channel=DBAGenesis
https://www.youtube.com/watch?v=8dA6BYxF_YE&ab_channel=jbleistein11
<https://www.techtarget.com/searchdatabackup/definition/Oracle-RMAN-Oracle-Recovery-Manager>
https://docs.oracle.com/cd/E11882_01/rac.112/e41960/admcon.htm#RACAD1111
<https://www.educba.com/what-is-oracle-rac/>
<https://www.thegeeksearch.com/beginners-guide-to-oracle-rac-architecture/>
<https://www.bing.com/videos/search?q=oracle+rac+architecture&docid=607998401175884580&mid=C35C67299013F6704336C35C67299013F6704336&view=detail&FORM=VIRE>
<https://oracle-patches.com/en/databases/oracle/oracle-rac-components>