

Univerzitet u Nišu, Elektronski fakultet

Katedra za računarstvo

Niš, 2021

Danilo Vulović

Seminarski rad

Microsoft SQL Server High Availability

**Mentor**: **Student**:

Prof. dr Aleksandar Stanimirović Danilo Vulović 1065

#### Sadržaj

[1. Uvod 4](#_Toc75729761)

[2. Visoka dostupnost 6](#_Toc75729762)

[2.1. Uzroci zastoja 6](#_Toc75729763)

[2.2. Metrike visoke dostupnosti 7](#_Toc75729764)

[2.2.1. Srednje vreme između otkaza – MTBF 8](#_Toc75729765)

[2.2.2. Srednje vreme do ozkaza – MTTF 8](#_Toc75729766)

[2.2.3. Srednje vreme popravak – MTTR 9](#_Toc75729767)

[2.2.4. Srednje vreme oporavka – MTTR 9](#_Toc75729768)

[2.2.5. Srednje vreme odgovora – MTTR 10](#_Toc75729769)

[2.2.6. Srednje vreme rešavanja – MTTR 10](#_Toc75729770)

[2.2.7. Srednje vreme potvrde – MTTA 10](#_Toc75729771)

[2.2.8. Metrika devetki 10](#_Toc75729772)

[3. MS SQL Server 11](#_Toc75729773)

[3.1. Pribavljanje podataka u MS SQL Serveru 11](#_Toc75729774)

[3.2. Servisi MS SQL Servera 11](#_Toc75729775)

[4. MS SQL Server HA rešenja 13](#_Toc75729776)

[4.1. Replikacija baza podataka 13](#_Toc75729777)

[4.1.1. Replikacija u MS SQL Serveru 14](#_Toc75729778)

[4.1.2. Replikacioni procesi u MS SQL Server-u 15](#_Toc75729779)

[4.2. Mirroring 16](#_Toc75729780)

[4.3. Always On Failover Cluster 17](#_Toc75729781)

[4.4. Always On Availability Groups 19](#_Toc75729782)

[5. Zaključak 20](#_Toc75729783)

[6. Literatura 21](#_Toc75729784)

# Uvod

Spajanje poslovanja sa internetom je promenilo način na koji mnoge firme funkcionišu. Od poslovnih procesa se sada očekuju, pa i podrazumevaju, bolja povezanost, fleksibilnost i veća dostupnost podataka. Osnovni posao baze podataka je da obezbedi sigurnost podataka i visoku dostupnost podataka u slučaju greške. Da bi smo obezbedili visoku dostupnost podataka, neophodno je obezbediti visoku dostupnost čitavog sistema. Ono što moramo da razumemo je da sistem može prestati sa radom u bilo kom trenutku. Možda je u pitanju softverska greška, ili pak nestanak struje gde se server nalazi, ali greške su u svakom trenutku moguće.

Okruženja koja su konfigurisana da obezbede skoro stalnu dostupnost podataka se nazivaju okruženjima sa podacima visoke dostupnosti (High availability – u daljem tekstu **HA**). Ovakvi sistemi imaju višak hardvera i softvera koji omogućava da sistem i dalje opslužuje zahteve uprkos problemima. Dobro dizajniran sistem visoke dostupnosti nema nijedan deo sistema zbog čije greške će ceo sistem prestati sa radom. Svaki deo, bio on softverski ili „gvožđurija“, koji može da prestane sa radom zbog neke greške, ima rezervni deo istog tipa, spreman da ga zameni.

Uzmimo za primer sajt koji omogućava bilo kakvu vrstu kupovine, poput *AliExpress-*a, *Amazon-*a itd. Ovi sajtovi su podložni iznenadnim povećanjima saobraćaja (npr. za vreme Crnog Petka) i nedostupnost od čak i nekoliko sekundi može da rezultira neuspešnim kupovinama, gubitku novca kao i narušenoj reputaciji čitavog brenda. Investicija u HA sistem će eliminisati bilo kakvu brigu o tome da li će sistem biti spreman za potencijalu navalu kupaca.

Još jedan primer gde je dostupnost podataka od krucijalnog značaja su banke. Od bilo koje finansijske institucije se očekuje da portal ili aplikacija bude konstantno dostupna. Ukoliko je offline period predugačak, korisnici će možda pomisliti da postoji razlog za paniku vezan za sigurnost njihovih podataka i novca. To će ultimativno dovesti do promene banke koju koriste. Iz tog razloga, ima smisla prebaciti čitavu infrastrukturu na visoko dostupan web-hosting.

Razmislimo o još jednom primeru kako bi lakše razumeli zašto je visoka dostupnost bitna. Taksi služba mora da poseduje vezu između svojih centara, vozača i korisnika „24/7“. Ako recimo, aplikacija prestane sa radom, putnici neće moći da naruče prevoz, vozači ne bi znali gde treba da idu za narednu mušteriju a sama služba bi izgubila reputaciju, a zaradu i da ne pominjemo. U ovom slučaju, ulaganje u HA sistem ne predstavlja luksuz, već zahtev.

Klasteri visoke dostupnosti podrazumevaju korišćenje većeg broja web servera, odnosno čvorova, koji će zajedno da se pobrinu da se vreme zastoja približi nuli ukoliko dođe do problema u nekom čvoru klastera. Kada dođe do otkaza nekog čvora, pokreće se proces koji operaciju koja nije uspela, izvršava na nekom drugom čvoru istog tipa. Ovaj proces se naziva „failover“ proces i on, posle otkaza nekog čvora, reorganizuje sistemske resurse, oporavlja neuspele transakcije i vraća sistem u normalan režim rada, i sve to, vrlo često, za vreme reda mikrosekundi. Ovaj proces ne treba da bude vidljiv korisniku. Korisnik je vrlo često za vreme iste sesije, uslužen od strane različitih čvorova bez njegovog znanja.

Naravno, iz svega ovoga, treba se zapitati, ko je ta osoba koja ne bi želela robustnu i pouzdanu infrastrukturu? Čak i vlasnici blogova ili sajtova koji ne očekuju veliki broj korisnika na dnevnom nivou nisu najsrećniji kada im je sajt nedostupan nekoliko minuta ili sati. Praktično, ulaganje u HA sistem može da bude izuzetno skupo, svaka gore pomenuta, rezervna komponenta, bilo softverska ili hardverska iziskuje dodatni trošak. Za mnogo okruženja važi da je cena sistema sa dva čvora dvostruko veća nego sa jednim čvorom isto kao što je i cena sistema sa četiri čvora dvostruko skuplja od sistema sa dva čvora. Takođe, povećan broj čvorova podrazumeva veću kompleksnost praćenja rada sistema kao i složeniju orkestraciju. Rezultat ovoga je da ne može svaki biznis da izbalansira troškove korišćenja klastera i prednosti smanjenog vremena zastoja. Najbolji način evaluacije da li je HA sistem za specifičan slučaj korišćenja isplativ je da odredimo učestalost i cenu zastoja sistema.

Za korisnike postoje mnogobrojna već gotova i spremna rešenja koja omogućavaju da njihova aplikacija bude dostupna konstantno kao i da se bilo kakvo opterećenje raspodeli po serverima kako ne bi došlo do gubitka performansi. *MySQL InnoDB Cluster* je MySQL rešenje koje nudi automatsku sinhronizovanu replikaciju, bezbedne transakcije i ugrađen proces za failover. Oracle, jedna od najkorišćenijih baza podataka, nudi nekoliko proizvoda i usluga koji obezbeđuju visoku dostupnost. Neki od njih su *Real Application Clusters*, *Oracle Real Application Clusters Guard*, *Oracle Replication* i *Oracle9i Data Guard*. Kombinacija ovih proizvoda se može koristiti da se namiri potreba određenog projekta za visokom dostupnošću podataka.

U ostatku rada, bavićemo se samim pojmom HA, kako se dostupnost meri kao i koji su najčešći razlozi za zastoj sistema (drugo poglavlje), kratak uvod u to šta je Microsoft SQL Server (treće poglavlje), koja HA rešenja nudi Microsoft SQL server (četvrto poglavlje) kao i da odgovorimo na pitanje kada i na koji način treba koristiti neko od HA rešenja (peto poglavlje). Poslednje poglavlje sadrži literaturu koju sam upotrebio za pisanje ovog rada (šesto poglavlje).

# Visoka dostupnost

Kao što smo već pomenuli, visoka dostupnost podrazumeva da je sistem u stanju da opsluži zahteve korisnika čak i pored grešaka koje mogu pogoditi sistem. Trebamo se zapitati, kako se dostupnost zapravo meri. Prvo što nam naravno pada na pamet je, meriti vremensku dostupnost servisa koji evaluiramo. To je, naravno, tačno ali je bitno izmeriti i očekivanu dostupnost određenog tipa sistema. Od imejl servera se očekuje mnogo veća dostupnost od aplikacija koje očekuju mali broj korisnika dok recimo, sistemi koji prate promene na berzi, zahtevaju kontinualnu dostupnost samo u određenim intervalima, odnosno, kada je berza otvorena. Danas je važnije nego ikada reševati kvarove sistema u što manjem vremenskom opsegu jer su posledice stvarne i nekad jako velike. Propušteni rokovi, kašnjenje uplata, neupeli projekti. Kompanije moraju da kvantifikuju i prate različite metrike koje se odnose na vreme rada, vreme zastoja i brzinu rešavanja problema.

## Uzroci zastoja

Zastoje pre svega klasifikujemo kao *planirane* i *neplanirane* zastoje. Planirani zastoji generalno odnose na zastoje izazvane sistemskim greškama, greške sa podacima ili medijumima i greške vezane za lokaciju samog servera.

Diagram

Description automatically generated

Slika 1 – Vrste neplaniranih zastoja

Neplanirani zastoji su problematičniji jer je nekada vrlo teško predvideti njihovu učestalost i ozbiljnost.   
 Sa druge strane, planirano zastoji mogu jednako da ometaju poslovanje, posebnu u globalnim sistemima koje koriste korisnici iz različitih vremenskih zona. Generalno se mogu klasifikovati kao rutinske operacije, periodično održavanje i nadogradnja sistema.

Diagram

Description automatically generated

Slika 2 – Planirani zastoji.

U ovom radu ćemo više pominjati neplanirane zastoje i kako umanjiti njihov negativni efekat korišćenjem različitih HA koncepata.

## Metrike visoke dostupnosti

Softverska industrija uglavnom koristi nekoliko metrika za

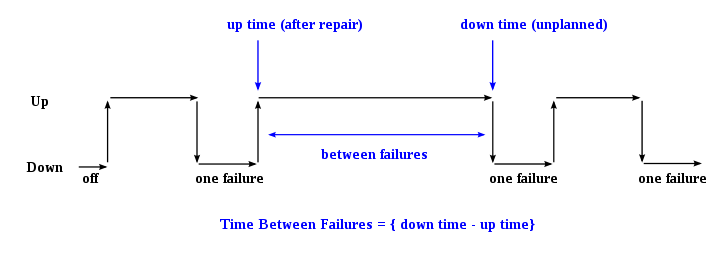
* Srednje vreme između otkaza *(The mean time between failures - MTBF)*
* Srednje vreme do otkaza (*The mean time to failure – MTTF)*
* Srednje vreme popravka (*The mean time to repair* – *MTTR1)*
* Srednje vreme oporavka (*The mean time to recovery* – *MTTR**[[1]](#footnote-1))*
* Srednje vreme odgovora (*The mean time to respond* – *MTTR1)*
* Srednje vreme rešavanja (*The mean time to resolve* – *MTTR1)*
* Srednje vreme potvrde (*The mean time to acknowledge – MTTA)*
* *Metrika devetki (Number of nines availability)*
* *…*

Ove metrike pomažu timovima da otkriju koliko se incidenti često dešavaju i koliko je potrebno da se sistem vrati u normalan režim rada. Naravno, treba ih uzeti sa određenom rezervom. One ne odgovaraju na bitnija pitanja poput, zašto se incident dogodio, šta od sistemskih komponenata i dalje radi a šta ne itd. Umesto toga, ove metrike nude dobru osnovu za pokretanje mnogo dubljih i ozbiljnijih pitanja. Za svaku od metrika trebamo postaviti nekoliko pitanja, kako se meri i kako i gde se koristi.

### Srednje vreme između otkaza – MTBF

Srednje vreme između otkaza je prosečno vreme između dva popravljiva otkaza nekog proizvoda. Koristi se da opiše dostupnost i pouzdanost sistema. Veće vreme između kvara, veća pouzdanost sistema.

Računa se korišćenjem aritmetičke sredine. U osnovi, to znači uzimanje podataka iz perioda za koji želimo da sračunamo vrednost (to može da bude npr. 1 godina ili 3 godine), i podeliti ukupno operativno vreme sistem za taj period sa brojem incidenata.

Slika 3 – MTBF (Srednje vreme između otkaza)

Recimo da je period koji nas zanima 48h. U tom periodu, bilo je 8h zastoja i 5 odvojena otkaza. To znači da je operativno vreme bilo 40 sati pa je samim tim MTBF jednak 40/8 = 5 sati.

Ova vrednost služi da pokaže kupcima koliko je proizvod pouzdan a programerima da bolje prate učestalost otkaza. Bitno je naglasiti da se MTBF koristi za sisteme koji su popravljivi. Ozbiljniji kvarovi koji zahtevaju promenu sistema, obično koriste srednje vreme do otkaza (MTTF).

### Srednje vreme do ozkaza – MTTF

Srednje vreme do otkaza je prosečno vreme između nepopravljivih ozkaza nekog produkta. Npr. ukoliko automobili neke marke mogu da voze 500 000 sati pre otkaza njihovih motora, MTTF tog proizvoda je 500 000. Najčešće se koristi da odredi životni vek proizvoda kao i da da naznaku kada nova verzija produkta daje veće performanse nego stara koja je na izmaku svog funkcionisanja. Još jedan primer za izračunavanje MTTF je izračunavanje životnog veka sijalica. Recimo da je uzet uzorak od 4[[2]](#footnote-2) sijalice i da su one imali životni vek od 22h, 25h, 26, i 19h, respektivno. MTTF se računa po formuli:

MTTF =

Zaključujemo da je MTTF za ovaj brend sijalica 92/4 = 23h.

Naravno, lako je „igrati se“ sa sijalicama. Njihov životni vek je relativno mali i lako merljiv. Šta se događa kada treba da se izmeri životni vek nečega što je zamišljeno da radi godiniama ili decenijama? Tada ova metrika, nije baš korisna jer nije jednostavno primenljiva. Slična je situacija sa softverom, jer je životni vek softvera zapravo kalkulacija do kada je verzija softvera koja se koristi isplativa.

### Srednje vreme popravka – MTTR

Srednje vreme za popravak je prosečno vreme potrebno za popravak sistema (obično tehničkog ili mehaničkog). Uključuje i vreme popravke i bilo koje vreme potrebno za analizu i ispitivanje ozkaza. Sat se zaustavlja tek kada je sistem potpuno funkcionalan. MTTR se može izračunati tako što ćete sabrati ukupno vreme provedeno u popravkama tokom bilo kog datog perioda i to vreme podeliti sa brojem popravki.

Dakle, recimo da pratimo popravke tokom jedne sedmice. U to vreme, bilo je 10 otkaza i na popravke tih otkaza je utrošeno 4 sata. Zaključuje se da je MTTR za ovaj slučaj 240 minuta /10 = 24 minuta.

Prosečno vreme popravke nije uvek isto vreme kao i sam prekid rada sistema. U nekim slučajevima popravke počinju u roku od nekoliko minuta nakon kvara proizvoda ili prekida rada sistema. U drugim slučajevima postoji vreme između problema, kada se problem otkrije i kada popravak započne. Ideja ove metrike je da prati koliko brzo osoblje za održavanje može da popravi kvar. Naravno, uvek postoji neko vreme između nastanka kvara i stizanja obaveštenja do osoblja koje radi održavanje.

### Srednje vreme oporavka – MTTR

Srednje vreme oporavka je prosečno vreme koje je potrebno za oporavak od kvara proizvoda ili sistema. Računa se deljenjem vremena provedenog u zastoju i broju kvarova za neki dat interval. Ako je neki naš servis u poslednja 24h, bio u zastoju 30 min ukupno u 2 odvojena incidenta, MTTR je jednak 30 minuta /2 = 15 minuta.

Kao što smo već pomenuli, put od kvara do oporavka nije jednostavan i ne sastoji se iz isključivo opravki. Srednje vreme oporavka može da identifikuje problem ali ne i da odgonetne gde je taj problem. Možda je problem u dijagnostici i greške se ne pronalaze brzo. Možda je problem u obaveštenjima i kvarovi se je prijavljuju dovoljno brzo po nastajanju. Možda tim za održavanje ne radi najefektivnije. Za odgovor na ova pitanja, mora se zakopati malo dublje. Možemo zaključiti da je srednje vreme oporavka mera brzine čitavog procesa oporavka.

### Srednje vreme odgovora – MTTR

Srednje vreme odgovora je prosečno vreme potrebno proizvodu ili sistemu od prvog trenutka od kada smo obavešteni o kvaru do potpunog oporavka. Kao što vidite, srednje vreme odgovora je neka vrsta podskupa srednjem vremenu oporavka. Slično se i računa s tim što se vreme potrebno da budemo obavešteni o kvaru ne uzima. Ova metrika se često koristi u sajber-bezbednosti prilikom merenja uspeha tima u neutralizaciji sistemskih napada.

### Srednje vreme rešavanja – MTTR

Srednje vreme rešavanja je prosečno vreme potrebno za potpuno rešavanje kvarova. To uključuje ne samo vreme utrošeno na otkrivanje kvara, dijagnostifikovanje i popravljanje već i vreme provedeno na osiguranju da se ovaj tip kvara više ne dogodi.

Ova metrika proširuje odgovornost tima za održavanje na dugoročno poboljšanje performansi. Velika je razlika između gašenja požara i gašenja požara a zatim postavljanje protivpožarnih aparata. Postoji jaka korelacija između srednjeg vremena rešavanja otkaza i satisfakcije klijenata. Računa se tako što se biznis sati potrebni za potpuno rešavanje incidenta podele sa brojem incidenata.

### Srednje vreme potvrde – MTTA

Srednje vreme potvrde se računa kao prosečno vreme potrebno da se počne sa radom na rešavanju problema od momenta kada je obaveštenje o otkazu pristiglo. Ovo metrika pomaže da se oceni responzivnost tima za održavanje kao i sistema za obaveštenja o ozkazima. Ova metrika je bitnija za internu upotrebu programera jer govori i tome da li tim za održavanje pati od prenatrpanosti i da li rešava probleme dovoljno brzo.

### Metrika devetki

Ova metrika je krajnje jednostavna i jednostavno se računa. Ukoliko je sistem bio nedostupan 526 minuta tokom čitave godine, to znači da je bio dostupan 99.9% vremena ili je ispunio dostupnost koju „diktiraju“ *tri devetke.* Ukoliko je sistem bio nedostupan samo 5 minuta na godišnjem nivou, ovakav sistem ispunjava dostupnost *pet devetke* itd. Ovu metriku je teško shvatiti ozbiljno bez prethodnog znanja o strogim procesima za upravljanje okruženjima, metodologija testiranja i procedurama oko sistemskih promena.

# MS SQL Server

Microsoft SQL Server je relacioni sistem upravljanja bazama podataka koji je razvio Microsoft. Kao server baze podataka, to je softverski proizvod sa primarnom funkcijom čuvanja i preuzimanja podataka na zahtev drugih softverskih aplikacija - koji se može pokretati ili na istom računaru ili na drugom računaru u mreži (uključujući Internet). Microsoft prodaje najmanje desetak različitih izdanja Microsoft SQL Server-a, namenjenih različitoj publici i za različita radna opterećenja, od malih pojedinačnih aplikacija do velikih aplikacija usmerenih ka Cloud-u sa mnogo istovremenih korisnika. Neke od edicija SQL servera su Azure, Azure SQL Data Warehouse, Embedded, Fast Track, Analytics Platform Systems itd. Takođe, SQL server dolazi u nekoliko različitih cenovnih opsega za različite korisnike. Enterprise je najskuplja edicija i nudi sve funkcionalnosti i sve dodatne alatke koje dolaze uz SQL server, dok su Standard i Express (besplatna verzija) verzije koje imaju manji skup funkcionalnosti. Najnovija bitna verzija MS SQL Servera je objavljena 11. aprila 2019. (15.x) godine sa nizom manjih nadogradnji objavljenih naknadno.

## Pribavljanje podataka u MS SQL Serveru

Glavni način preuzimanja podataka iz baze podataka MS SQL Servera je postavljanje upita. Upit se izražava pomoću varijante SQL-a koja se zove T-SQL, dijalekta koji Microsoft SQL Server deli sa Sybase SQL Serverom. Upit deklarativno navodi šta treba da se preuzme. Obrađuje ga procesor upita, koji utvrđuje redosled koraka koji će biti potrebni za preuzimanje traženih podataka. Redosled radnji neophodnih za izvršavanje upita naziva se plan upita. Vrlo često postoji više načina za obradu istog upita. Na primer, za upit koji sadrži izraz za pridruživanje i izraz za odabir, izvršavanje pridruživanja na obe tabele, a zatim izvršavanje izbora na rezultatima daće isti rezultat kao i odabir iz svake tabele, a zatim izvršavanje pridruživanja, ali rezultira drugačijim planom izvršenja. U tom slučaju, SQL Server bira plan za koji se očekuje da će dati rezultate u najkraćem mogućem roku. To se naziva optimizacija upita i vrši je sam procesor upita.

## Servisi MS SQL Servera

SQL Server takođe uključuje asortiman dodatnih usluga. Iako one nisu ključne za rad sistema baza podataka, one pružaju usluge sa dodatom vrednošću povrh osnovnog sistema upravljanja bazama podataka. Ove usluge se pokreću ili kao deo neke SQL Server komponente ili su van procesa kao Windovs servis i predstavljaju sopstveni API za kontrolu i interakciju sa njima. Neki od najbitnijih servisa su:

* Service Broker
* Servisi za mašinsko učenje
* **Servisi za replikaciju**
* Servisi za obaveštavanje
* Servisi za izveštavanje
* Servisi za integraciju
* Servisi za potpunu pretragu teksta,
* SQL Server Management Studio – jednostavan GUI za sve operacije nad SQL Server-om u upravljanje istim kao i dodatnim alatima
* Visual Studio
* Itd.

U narednom poglavlju ćemo se pozabaviti nekim od ovih servisa i objasnićemo njihovu vezu u implementaciji nekog HA koncepta u MS SQL Serveru.

# MS SQL Server HA rešenja

Svaka od HA metoda dostupnih u Microsoft SQL serveru ima svoje jedinstveno mesto i pruža određeni nivo fleksibilnosti. Klasteri baza podataka su izuzetno korisni za svakoga, ali nisu svima potrebni. Na osnovu potreba sistema, treba oceniti koji HA metod će doneti najmanje novčane gubitke kada se uzme u obzir cena zastoja kao i cena primene HA metoda.

## Replikacija baza podataka

Replikacija baza podataka je proces učestalog kopiranja podataka iz baze podataka sa jednog servera u bazu nekog drugog servera tako da svi korisnici imaju pristup istom nivou informacija. Rezultat ovoga je distribuirana baza podataka gde svaki korisnik može jednostavno pristupiti podacima koji su potrebni njegovom poslovanju bez ometanja drugih korisnika. Brojni elementi doprinose ukupnom procesu stvaranja i upravljanja replikacijom baze podataka. Distribuirani sistem koji se brine o svim bazama i čitavom procesu disribucije i replikacije se zove distribuirani sistem za upravljanje bazama podataka (DDBMS). On se brine da bilo kakva promene, dodavanja podataka ili brisanje na bilo kojoj lokaciji, bude reflektovane i na podacima sa drugih lokacija. Dakle, DDBMS brine o distribuiranim baza podataka koje su proizvod replikacije podataka. Klasičan primer ovakvih replikacija je jedna ili više aplikacija koja ima primarnu bazu i sekundarnu bazu koja se nalazi na drugoj lokaciji kako se greške sa prvom bazom, ne bi prenele i na drugu.

Diagram

Description automatically generated

Slika 4 – Replikacija

Na slici 4 se vide osnovni akteri za vreme procesa replikacije. **Publisher** je server koji nudi podatke drugim serverima. Takođe, za vreme procesa sinhronizacije, publisher računa razlike u podacima. **Subscriber** je server koji prima podatke i čuva podatke koji su objavljeni od strane publisher-a. Izmene podataka kod subscriber-a, mogu biti prenete nazad kod publishera u zavisnosti od podešavanja. **Distributer** je server koji upravlja protokom podataka kroz proces replikacije. Prisutna su dva tipa distributera, jedan je udaljen, a drugi lokalni distributer. Udaljeni distributer je odvojen od publisher-a i konfigurisan je kao distributer za replikaciju. Lokalni distributer je server koji je konfigurisan kao publisher i distributer.

### Replikacija u MS SQL Serveru

MS SQL Server nudi nekoliko vrsta replikacija za korišćenje u distribuiranim aplikacijama:

* Replikacija na nivou transakcije

Promene na publisher-u[[3]](#footnote-3) se dostavljaju subscriber-u[[4]](#footnote-4) u približno realnom vremenu i primenjuju se u istom redosledu i u istom transakcionom opsegu kao na Publisher-u.

* Replikacija zasnovana na spajanju (merge)

Promene se mogu dešavati i na publisheru i na subscriber-ima i konstantno su praćene. Svaki put kada se subscriber sinhronizuje sa publisher-om, razlika u redovima se otkriva od poslednje sinhronizacije i ažuriraju se.

* Replikacija snapshot-ovima

Slika svih podataka publisher-a (koja se naziva *snapshot*) se šalje subscriber-ima periodično i izvršava sinhronizaciju svojih i pristiglih podataka. Prednost ove metode jer nema strogog praćenja promena baza podataka.

* Peer-to-peer replikacija

Slična filozofija kao replikacija na nivou transakcija ali nema primarnog servera. Sve baze se smatraju jednakim članovima i promene na jednoj se šalju svim ostalim instancama.

* Bidirekciona replikacija

Ova replikacija ne garantuje postojanje primarne baze podataka već garantuje postojanje mehanizma pomoću kog se dva servera sinhronizuju tako što jedan server objavi svoje promene za ostale servere a zatim čeka promene sa ostalih servera.

* Replikacija subscriber-ima

Promene na bilo kom subscriber-u se propagiraju prvo publisher-u a on šalje svim ostalim subscriber-ima.

Vrsta replikacije koja se koristi zavisi od mnogo faktora, kao što su fizički medijum koji se koristi za bazu podataka, vrsta i kvantitet podataka koji se replicira i jako je bitno pitanje da li subscriber-i mogu slati promene ili ne. Fizičko okruženje replikacije uključuje broj i lokaciju računara koji se koriste kao i vrsta klijenta (radna stanica, laptop, mobilni telefoni itd.).

### Replikacioni procesi u MS SQL Server-u

Rekli smo već da replikacija kao process predstavlja rad više elemenata koji zajedno čine process replikacije moguć. MS SQL Server koristi nekoliko zasebnih programa koji se nazivaju *agenti* koji izvršavaju zadatke koji su povezani sa praćenjem promena i distribuiranjem podataka. Postoji jedan glavni agent koji se zove *SQL Server Agent* i svi ostali agenti zahtevaju da on funkcioniše. Neki od agenata koji rade su:

* Snapshot Agent
  + Ovaj process se tipično koristi u svim tipovima replikacije jer priprema šemu i inicijalnu publikaciju. On se izvršava u distributeru.
* Log Reader Agent
  + Ovaj agent se koristi sa replikacijom transakcija. Premešta transakcije označene za replikaciju iz dnevnika transakcija na Publisher-u u bazu podataka distribucije. Svaka baza podataka objavljena korišćenjem transakcione replikacije ima svog agenta čitača dnevnika koji radi na distributeru i povezuje se sa izdavačem
* Distribution Agent
  + Distributivni agent se koristi sa replikacijom snapshot-a i transakcijskom replikacijom. Primenjuje inicijalni snapshot na Subscriber-u i premešta transakcije koje se čuvaju u distribucijskoj bazi podataka na Subscriber-e.
* Merge Agent
  + Ovaj agent se koristi sa replikacijom spajanja. Primenjuje početni snapshot na subscriber-a i pomera i usklađuje inkrementalne promene podataka koje se javljaju. Svaka pretplata na spajanje ima svog agenta za objedinjavanje koji se povezuje i sa publisherom i sa subscriberom i ažurira oba.
* Queue Reader Agent
  + Agent za čitanje iz reda koristi se sa replikacijom transakcija sa opcijom ažuriranja u redu. Agent radi kod distributera i vraća izmene izvršene kod subscribera na publishera. Za razliku od distribucijskog agenta i agenta objedinjavanja, postoji samo jedan primerak agenta čitača iz redova koji servisira sve publishere i publikacije za datu bazu podataka distribucije.
* Replication Maintenance Agents

## Mirroring

Funkcioniše tako što održava dve kopije jedne baze podataka koje se nalaze u različitim instancama SQL Servera. Oni su često rašireni po različitim data centrima ili geografskim lokacijama kako bi se zaštitili od kvarova i svega što „gori do temelja”,

Jedna od dve instance opslužuje zahteve iz aplikacije, dok druga instanca čeka u stanju pripravnosti, spremna da preuzme posao u trenutku.

Pokretanje ovih baza podataka u režimu visoke sigurnosti osigurava da su podaci uvek potpuno isti u obe baze podataka, pri čemu se transakcije istovremeno zapisuju u obe baze podataka pre nego što se odgovor može poslati nazad na PaperCut.

Mirrorring takođe podržava failover bez gubitka podataka putem „svedoka” servera. Ovo je zaseban Microsoft SQL server koji nadgleda primarnu instancu i osigurava nesmetan prelazak u slučaju prekida. Ako se instance međusobno odvoje, oni se oslanjaju na svedoka kako bi bili sigurni da samo jedan od njih trenutno komunicira sa aplikacijom.

Diagram

Description automatically generated

Slika 5 – Mirroring u MS SQL-u sa svedok serverom

Imati tačne kopije baze podataka uvek dostupne i na mreži omogućava kupcu da ručno prebaci sistem - a da aplikacija nema nikakvo znanje o promeni instance koja radi. Možda se čini savršeno rešenje, ali trenutno je u režimu održavanja i moglo bi da se ukloni u budućoj verziji Microsoft SQL Servera. Microsoft preporučuje prebacivanje na *Availability Groups* i nudi alate koji pomažu u prelasku sa Mirroring tehnike.

## Always On Failover Cluster

Aplikacija se sastoji od dva dela: delova u RAM-u (sve što je keširano i svi tekući upiti) i čvrstih diskova (baze podataka). Klasteri rade nešto prilično sjajno: uzimaju nešto poput korisničkog SQL servera i odvajaju ga od hardvera na kojem radi. Ako bi kupac mogao da uzme te diskove i deli ih sa više računara, onda bi mogao da uzme svoj SQL server i da ga lako postavi na novi hardver u trenutku. I upravo to vam omogućavaju klasteri.

Sa klaster okruženjem možete da konfigurišete više servera da koriste isti skup deljenih diskova. Samo jedan server će istovremeno koristiti tu memoriju.

Ako kupac ima kvar na komponenti, a taj fizički server propadne, ima u pripravnosti ceo drugi fizički server da zgrabi deljene diskove i krene na posao. A to rezultira kratkim zastojima.

Pored toga što su izuzetno zgodni za kvarove, klasteri su zaista odlični i za redovno održavanje. Na primer, recimo da izlazi novi servisni paket za SQL i da kupac želi da ga primeni. Da bi to uradili, oni moraju da zaustave uslugu i preuzmu baze podataka van mreže. Ali šta će se desiti sa njihovim krajnjim korisnicima kada to urade? Neće biti u mogućnosti da koriste njihove usluge. Da bi to izbegli, zakazuju ažuriranje aplikacije u 3 ujutru u subotu što nije baš zgodno. Ali pomoću klastera, kupac to sve može zaobići - zahvaljujući dva ili više servera na raspolaganju.

Diagram

Description automatically generated

Slika 6 – MS SQL Server failover klaster

Dakle, server A može da pokreće SQL, a server B – koji samo čeka u stanju pripravnosti, ne radeći puno - može na sebe primeniti servisni paket. Tada kupci mogu da uzmu svoj SQL server, isključe ga i primene na njega ažuriranje. Oba servera su ažurirana i krajnji korisnici nikada nisu izbačeni.

Sve je u redu sa osvim ali najveći nedostatak klastera je trošak oko njih. Drugi server košta duplo - a i nešto preko toga - jer diskovi moraju da se dele između dva servera. Vrlo često, za korisnike je jeftnije da prihvate određen period zastoja za vreme ažuriranje SQL Servera umesto što plaćaju konstantnu povezanost dva ili više servera.

Osim toga, čak i sistemi koji imaju klaster, još uvek nisu imuni na otkaze. Svakako, mogao bi da propadne čitav jedan server - a budući da drugi server preuzima na sebe, ne moraju da brinu o procesoru, memoriji, matičnoj ploči, napajanju i svemu ostalom što bi moglo da zakaže. Ali određene nepogode koje mogu da se dogode poput požara mogu da unište sve čvorove u klasteru a velika geografska distanca, koja bi i ovaj problem zaobišla, za klastere po prirodi nije dobra jer zahtevaju komunikaciju u realnom vremenu. Postoje tehnologije koje će vam to omogućiti poput geoistezanja ali tu je uključeno kašnjenje, što nije dobro za SQL Server. U tom slučaju je za takve kupce bolje rešenje neke od drugih metoda, preslikavanje ili korišćenje grupa dostupnosti.

## Always On Availability Groups

Grupe dostupnosti su HA rešenje za oporavak od katastrofe koje pruža alternativu mirroring tehnici na nivou preduzeća. Kao i preusmeravanje klastera, to zavisi od Windovs klastera, tako da ima jednu nepromenljivu tačku za povezivanje aplikacije dok se baze podataka pomeraju iza zavese.

Za razliku od failover klasterizacije, grupama dostupnosti nije potrebno deljeno skladište, što je posebno važno ako kupac želi da bude host u Cloud-u, a ušteda troškova bi trebalo da udovolji njihovom finansijskom direktoru.

Takođe, ne rade na nivou SQL instance. Umesto toga, oni prelaze direktno na pojedinačne baze podataka, pa se samo baza podataka aplikacije kreće između SQL instanci u slučaju kvara.

Grupe dostupnosti repliciraju podatke slično kao i Mirroring. Kupci mogu znati da se njihova brza SSD memorija u skoro realnom vremenu replicira na spremni rezervni server u blizini i na drugi u potpuno drugoj zemlji ili regionu. Pored toga, oni su u mogućnosti da ručno pređu na failover za ažuriranja.

Negativna strana je postavljanje i održavanje grupa dostupnosti nije samo komplikovano - dolazi po značajnu cenu zbog upotrebe Microsoft SQL Server Enterprisea i zahteva od kupca da ima jedan ili više sekundarnih servera na mreži i u praznom hodu, čekajući da se dogodi neuspeh.

Dakle, osim ako specifični slučajevi upotrebe koje grupe dostupnosti pružaju nisu kritični - a cena pristupačna - ulaganje i rizik za većinu kupaca jednostavno ne vrede.

# Zaključak

Može biti zastrašujuće započeti potragu za rešenjem za replikaciju podataka koje će raditi za vaše posebne potrebe. Ali pronalaženje tog rešenja znatno će olakšati sve procese. Vaše IT odeljenje može samostalno da napiše kod i bavi se postupkom replikacije, ali ovo predstavlja niz sopstvenih poteškoća. Moraćete da posvetite vreme održavanju podataka, trošite novac na aplikacije i možda čak unajmite nekoliko dodatnih ljudi koji će pojednostaviti postupak. Osim toga, morate biti svesni zastrašujuće pretnje ljudske greške.

Zbog toga su replikacija podataka i sigurnosna kopija baze podataka toliko korisni. Rešenja za rezervne kopije baza podataka pomažu preduzećima da zaštite svoje podatke rezervnim kopijama u slučaju oštećenja podataka, korisničke greške ili hardverskog kvara. Koristeći rešenja za pravljenje rezervnih kopija baza podataka, preduzeća mogu osigurati da su njihovi podaci uvek dostupni, čak i ako njihova glavna baza podataka zakaže.

SQL Server je softverski paket unapređivan godinama i nudi širok asortiman mogućnosti za različite nivoe replikacija a na klijentima je da isplaniraju i odrede koji paket se najviše isplati, logistički i, možda još bitnije, novčano.

# Literatura

1. Bertucci, B, 2004, *Microsoft SQL Server High Availability*, Sams
2. Rankins, R, Bertucci, P, Gallelli, C, Silverstein, Alex, T, June 2015, *Microsoft SQL Server 2014 Unleashed*, Sams
3. Isakov, V, 2017, *Administering a SQL Database Infrastructure,* Microsoft Press
4. Jorgensen, A, Segarra, J, LeBlanc, P, Chinchilla, J, Nelson, A, August 2012, *Microsoft SQL Server 2012 Bible,* Wiley
5. Osama, A, July 2018, *Professional Azure SQL Database Administration*, Packt Publishing
6. Osama, A, January 2019, *Professional SQL Server High Availability and Disaster Recovery,* Packt Publishing

1. Primetimo da je skraćenica ista za nekoliko metrika, vrlo je bitno precizno definisati o kojoj metrici se priča jer se one delimično preklapaju ali fundamentalno govore o različitim stvarima [↑](#footnote-ref-1)
2. U realnosti, koristi se mnogo veći broj jedinica za testiranje kako bi se dobio što bolji statistički podatak [↑](#footnote-ref-2)
3. Publisher predstavlja primarnu bazu podataka koju aplikacija koristi. [↑](#footnote-ref-3)
4. Subscriber je sekundarna kopija baze podataka. [↑](#footnote-ref-4)