



Sistemi za upravljanje bazama podataka

Klaster rešenja - SQL Server

Mentor:

doc. dr Aleksandar Stanimirović

Student:

Natalija Mitić, 1046

Sadržaj

Uvod.....	3
Klasteri kod DBMS-ova	3
Postavke i razlozi kreiranja klastera	3
Funkcionisanje arhitekture klastera	5
Tipovi klastera baza podataka.....	5
SQL Server.....	8
Failover klaster.....	8
Šta SQL Server failover klaster obezbeđuje?	9
Zašto je SQL klasterovanje korisno?	11
Šta klasterovanje ne obuhvata?	12
SQL Server failover klaster konfiguracije	12
Kvorum	13
Ponovno pokretanje klastera	17
Kreiranje klastera	18
Zaključak.....	24
Literatura.....	24

Uvod

Baza podataka je server koji može da čuva, filtrira i pretražuje podatke. Ponekad jedan server nije dovoljan da obradi određenu količinu podataka ili zahteva. Tada je potreban klaster baza podataka. Klasterovanje baze podataka predstavlja postojanje više računara koji rade zajedno i koji se zajedno koriste za skladištenje podataka. Oni se udružuju kako bi pružili uslugu.

Postoji mnogo načina grupisanja baza podataka i često se kombinuju da bi se rešili konkretni slučajevi upotrebe.

Uprkos svim distribucijama ove tehnologije na *back-endu*, korisniku se čini da interaguje sa jedinstvenim sistemom. Upotreba klastera varira od organizacije do organizacije, zavisno od vrste procesa i nivoa potrebnih performansi.

U ovom radu je opisano funkcionisanje klastera i prikazano je šta klasteri omogućavaju. Na kraju su dati primeri postupka kreiranja klastera SQL Servera i prebacivanje resursa sa jednog na drugi čvor.

Klasteri kod DBMS-ova

Klasterovanje baze podataka se odnosi na mogućnost više servera ili instanci da se povežu u jedinstvenu bazu podataka. Instanca je kolekcija memorije i procesa koji interaguje sa bazom podataka koja predstavlja skup fizičkih datoteka koje zapravo čuvaju podatke.

Iz perspektive baze podataka, klasterovanje je kada postoji grupa mašina (čvorova) koji hostuju istu šemu baze podataka na istom softveru baze podataka sa nekim oblikom razmene podataka između ovih mašina. Primenjuje se tamo gde se računarska snaga može distribuirati na više hostova horizontalnim skaliranjem. Izvan klastera, ove mašine se posmatraju kao jedna celina koja sadrži uniju podataka koji su raspoređeni po čvorovima u klasteru. Kad je opterećenje previsoko, poželjno je podeliti (distribuirati) to opterećenje na više čvorova. Budući da bi previše pretraživanja moglo trajati duži vremenski period, potrebno je da se delovi podataka čuvaju u različitim čvorovima. Obično korisnik sam bira kako se podaci distribuiraju između čvorova, a pošto se zna koji su podaci sadržani u određenom čvoru, može se u kodu usmeriti na taj čvor, čineći pretragu baze podataka mnogo bržom. Kada aplikacija pristupa klasteru, zahtev se na kraju preusmerava na jedan čvor u klasteru za operaciju čitanja ili pisanja. Ti čvorovi nisu nužno replike, što znači da se podaci koji se čuvaju u svakom čvoru mogu razlikovati od ostalih čvorova, što razlikuje klaster od skupa replika. Klasterovanje štiti od grešaka u hardveru ili softveru servera, ali ne štiti od pada diska, zato što disk nije dupliciran.

Na ovaj način je omogućeno korišćenje više servera aplikacija za pristup istoj bazi podataka. Klasterovanje se koristi se za intenzivne računске operacije, paralelne i analitičke aplikacije koje rade na postojećim podacima.

Postavke i razlozi kreiranja klastera

Čvor je pojedinačna instanca baze podataka. Postoje dve vrste čvorova - *master* čvorovi i *slave* čvorovi. U tipičnom okruženju klastera, *master* čvor je nalik glavnom čvoru, a *slave*

čvorovi samo kopiraju podatke iz mastera. Iz perspektive aplikacije, odnosno korisnika, *master* čvor ima pristup čitanju i pisanju, a *slave* čvorovi imaju pristup samo za čitanje. Dakle, ako treba sačuvati podatke u bazu podataka, oni se prvo moraju sačuvati u *master* čvoru. Tada se ti podaci kopiraju u *slave* čvor i odatle se mogu pročitati.

U nekim su klasterima su svi čvorovi napravljeni kao *master* čvorovi. Ovo dovodi do dve različite topologije: *master-master* klasteri i *master-slave* klasteri. Primarna razlika između ove dve topologije je u tome što *master-master* klaster omogućava svakom čvoru da piše u bazu podataka, dok *master-slave* klaster omogućava samo određenim čvorovima (*master* čvorovima) da to urade. U podešavanjima kod kojih postoji mnogo zahteva za čitanje, a ne toliko zahteva za pisanje, podaci se mogu čitati sa bilo kog *slave* čvora, a ako je potrebno izvršiti upis, onda se to obavlja na *masteru*. Sve promene se zatim prosleđuju *slave* čvorovima. Ovo podešavanje je dobro za gusti saobraćaj čitanja. Ponekad je klasterovanje više vezano za sigurnost podataka. To znači da podaci moraju postojati na više servera, da bi se osiguralo da neće nestati zbog rušenja servera. Dakle, topologija *master-master* praktično znači da se dva ili više servera miroruju.

Glavni razlozi za klasterovanje baze podataka su njegove prednosti koje server dobija, a to su: Redundantnost podataka, balansirano opterećenje, velika dostupnost, kao i nadgledanje i automatizacija.

Redundantnost podataka (Data Redundancy)

Više računara sarađuje kako bi međusobno sačuvali podatke pomoću klasteriranja baza podataka. To daje prednost u vidu redundiranosti podataka. Pri tom, ovde se redundantnost ne odnosi na suvišnost podataka koja se dobija prilikom loše dizajnirane baze podataka. Razlog zbog kojeg redundantnost podataka nije problem u klasterovanju je taj što će korišćeni računari biti sinhronizovani. To znači da će svaki čvor imati iste podatke kao i svi ostali čvorovi, čak i ako se podaci promene. To zauzvrat znači da neće postojati jedan skup podataka koji sadrži određene vrednosti, a drugi skup podataka neke druge vrednosti. Sinhronizacija pomaže u sprečavanju loše vrste redundantnosti podataka.

U bazama podataka se teži da se izbegne redundantnost podataka koja vodi do dvosmislenosti podataka. Redundantnost koju klasterovanje nudi je izvesna zbog sinhronizacije. U slučaju da jedan računar ne radi, svi podaci će biti na raspolaganju na drugim računarima.

Balansirano opterećenje -skalabilnost (Load Balancing - scalability)

Balansiranje opterećenja ili skalabilnost ne dolazi podrazumevano sa bazom podataka. To se mora omogućiti redovnim klasterovanjem. Takođe, zavisi i od podešavanja. U osnovi, balansiranje opterećenja vrši distribuciju opterećenja između različitih računara koji su deo klastera. To ukazuje da se može podržati više korisnika i ako se iz nekog razloga pojavi ogroman skok u saobraćaju, postoji veća sigurnost da će novi saobraćaj biti podržan. Jedna mašina neće dobiti sve pogotke. Ovo može da obezbedi skaliranje bez problema, ako je potrebno, što se direktno povezuje sa velikom dostupnošću. Bez balansiranja opterećenja,

određena mašina mogla bi biti preopterećena i saobraćaj bi se usporio, što bi vodilo ka smanjenju saobraćaja.

Velika dostupnost (High Availability)

Kada se može pristupiti bazi podataka, to podrazumeva da je ona dostupna. Velika dostupnost odnosi se na količinu vremena kada se baza podataka smatra dostupnom. Na primer, ako je baza podataka dostupna samo 99 procenata vremena, nastaće određeni problemi, jer to znači da 3,6 dana u godini softver neće biti u mogućnosti da radi. Količina dostupnosti koja je potrebna korisniku uveliko zavisi od broj transakcija koje se vrše u bazi podataka i koliko često se koristi bilo koja vrstu analitike nad podacima. Pomoću klasterovanja baze podataka, može se dostići izuzetno visok nivo dostupnosti zbog dva glavna razloga:

- Balansiranje opterećenja - Bez balansiranja opterećenja, određena mašina bi mogla biti preopterećena i saobraćaj bi se usporio.
- Postojanje dodatnih mašina - Ako se kojim slučajem jedan server ugasi, baza podataka će i dalje biti dostupna.

Nadgledanje i automatizacija (Monitoring and Automation)

Za ovaj zadatak se može koristiti obična baza podataka, jer se nadzor i automatizacija mogu lako izvršiti pomoću softvera. Prednost postaje sve očiglednija kada je prisutan klaster. Tipično je prednost u tome što klasterovanje omogućava automatizaciju mnogih procesa baze podataka istovremeno kada se omogućava postavljanje pravila radi upozoravanja na potencijalne probleme. Ovo sprečava povratak kako bi se sve ručno proverilo. Kod klasterovane baze podataka automatizacija je korisna, jer omogućava dobijanje obaveštenja ako je sistem preopterećen. Međutim, klaster će imati određenu mašinu koja će se koristiti kao sistem za upravljanje bazama podataka - kontrolna tabla za ceo klaster. Ova odabrana mašina može imati skripte koje se automatski pokreću za čitav klaster baza podataka i rade sa svim čvorovima baze podataka.

Funkcionisanje arhitekture klastera

U arhitekturi klastera, svi zahtevi su podeljeni ka različitim računarima tako više računarskih sistema izvršava i obrađuje pojedinačni korisnički zahtev. Klasterovanje je definitivno korisno zahvaljujući mogućnosti balansiranja opterećenja i velike dostupnosti. Ako se jedan čvor sruši, zahtevom upravlja drugi čvor. Shodno tome, postoji malo ili nimalo mogućnosti apsolutnih kvarova sistema.

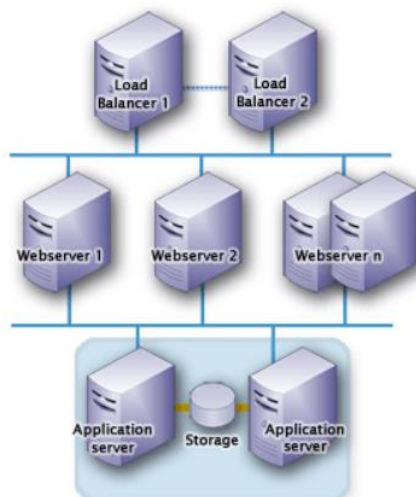
Tipovi klastera baza podataka

Klaster baza podataka je vrlo obiman. Obuhvata više nivoa, u zavisnosti od zahteva sistema. U nastavku su navedena tri najčešće arhitekture klastera.

Klasteri sa preuzimanjem posla instance koja je pala (Failover), odnosno klasteri sa velikom dostupnošću

Mašina može da se pokvari i da prestane sa radom u bilo kom trenutku. Administratori sistema upravljaju takvim padovima i efikasno rešavaju probleme. Ovde klasteri dolaze u pomoć. Klaster priprema raspoloživost usluge repliciranjem servera i rekonfiguracijom redundantnog softvera i hardvera. Dakle, svaki sistem kontroliše drugi i radi na zahtevima ako neki od čvorova padne. Ove vrste klastera su profitabilne za one korisnike koji u potpunosti zavise od svojih računarskih sistema (na primer, e-trgovina, web sajtovi itd). Sistem bi trebalo da bude dovoljno sposoban da zna koji se sistemi pokreću, s kojeg se IP-a pokreću, koji zahtev i kakav bi bio tok u slučaju pada. Važno je da serveri nikako ne bi trebalo da prestanu da rade.

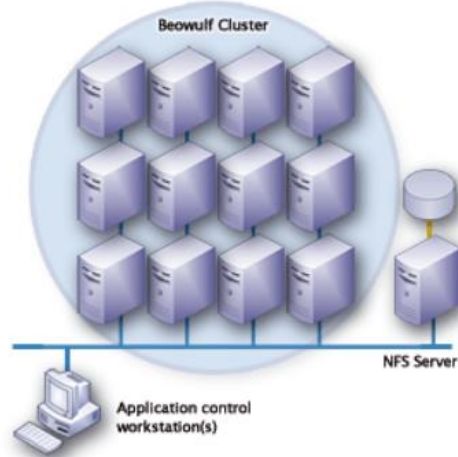
Ako jedan od čvorova u klasteru postane nedostupan, bilo zbog planiranog prekida održavanja ili zbog neplaniranog zastoja zbog kvara, drugi čvor preuzima kontrolu nad uslugom krajnjeg korisnika - proces poznat kao *failover*. Kada dođe do prebacivanja, korisnici koji pristupaju klsterskoj usluzi nastavljaju pristup usluzi i nisu svesni da se ona sada pruža s drugog servera (čvora). Ova arhitektura naglašava dostupnost baze podataka ili usluge aplikacije, a ne funkciju performansi ili balansiranje opterećenja.



Slika 1 – Failover klaster

Klasteri visokih performansi

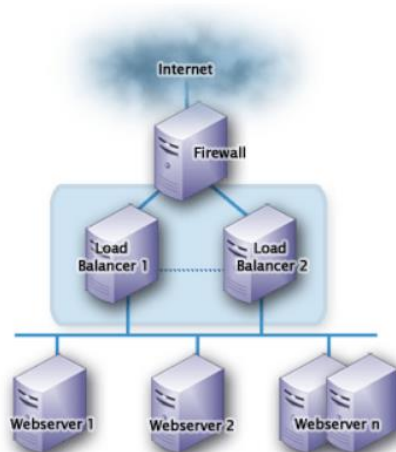
Svrha razvoja klastera baza podataka visokih performansi je izrada računarskih sistema visokih performansi, odnosno skaliranje računarske moći. Oni izvršavaju proširene programe koji su potrebni za izračunavanja koja zahtevaju dosta vremena. Takvu raznolikost klastera najčešće preferiraju naučne industrije. Osnovni cilj je inteligentna podela opterećenja. Da bi to bilo omogućeno, potrebno je podeliti posao na više „podposlova“. Jedinica za upravljanje se koristi za koordinaciju svih tih podposlova, a poseban čvor za rezultate je pojedinačna tačka pada. Pad jednog od računskih čvorova ne predstavlja katastrofu, jer bi posao mogao da radi drugi čvor. Performanse celog sistema bi bile ometene, ali ukupan učinak ne bi bio ugrožen.



Slika 2 – Klaster visokih performansi

Klasteri za balansiranje opterećenja

Farma servera sa istom funkcijom je osnova za balansiranje opterećenja. Ovi klasteri baza podataka služe za distribuciju opterećenja između različitih servera. Nastoje da obezbede povećan kapacitet mreže, konačno povećavajući performanse. Sistemi u ovoj mreži integrišu svoje čvorove uz pomoć kojih su zahtevi korisnika podeljeni podjednako po čvorovima koji učestvuju. Sistem ne radi zajedno, već preusmerava zahteve pojedinačno u trenutku kada se pojave. Čvor sa procenjenim najboljim performansama će dobiti sledeći korisnički zahtev za obradu. Ako jedan čvor nije u funkciji, onda će čitav sistem i dalje raditi. Balans opterećenja prepoznaće čvor koji je pao i obeležiće srušeni sistem. Ukupni učinak klastera za balansiranje opterećenja biće smanjen, ali usluge će i dalje biti pružene.



Slika 3 – Klaster za balansiranje opterećenja

SQL Server

Microsoft SQL Server klaster nije ništa drugo do skup dva ili više fizičkih servera sa identičnim pristupom deljenoj memoriji koja pruža diskovne resurse potrebne za smeštanje datoteka baze podataka.

Windows klasterovanje je opcija visoke dostupnosti koja je dizajnirana kako bi povećala radni vek instanci SQL Server-a. Jedan čvor je identifikovan kao aktivni čvor, na kome instance SQL Servera radi, a drugi je pasivni čvor, na kome je SQL Server instaliran, ali nije pokrenut. Ako instance SQL Servera na aktivnom čvoru ne uspe, pasivni čvor postaje aktivni čvor i započinje s radnim opterećenjem SQL Servera s minimalnim zastojem prebacivanja. Pored toga, Windows Failover klaster se može podesiti da ima više aktivnih čvorova, što znači da se pokreću različite instance SQL Server-a u kojima bilo koja instance SQL Servera može preći na drugi čvor.

Failover klaster

SQL Server failover klasteri su napravljeni od grupe servera koji pokreću aplikacije koje podržavaju klaster na takav način da umanjuju zastoj. Prebacivanje (eng. failover) je proces koji se dogodi ako se jedan čvor sruši ili postane nedostupan, a drugi preuzme i ponovo pokrene aplikaciju bez ljudske intervencije.

Failover klaster instance (FCI) se sastoji od skupa fizičkih servera (čvorova) koji sadrže sličnu konfiguraciju hardvera, kao i identičnu konfiguraciju softvera koja uključuje verziju operativnog sistema, kao i verziju SQL servera, komponente i naziv instance. Identična konfiguracija softvera je neophodna kako bi se osiguralo da klaster instance može biti u potpunosti funkcionalna, jer prelazi između čvorova.

FCI SQL Server se pokreće u WSFC (Windows Server Failover Cluster) grupi resursa. Svaki čvor u grupi resursa održava sinhronizovanu kopiju konfiguracionih postavki kako bi se osigurala puna funkcionalnost FCI-ja nakon prebacivanja, a samo jedan od čvorova u klasteru poseduje grupu resursa u isto vreme (aktivni čvor). Usluga WSFC upravlja klasterom servera, konfiguracijom kvoruma, politikom prelaska i operacijama prelaska, kao i virtuelnim IP adresama za FCI. U slučaju neuspeha (kvarovi hardvera, kvarovi operativnog sistema, kvarovi aplikacije ili usluge) ili planirana nadogradnja, vlasništvo nad grupom resursa se premešta na drugi čvor u FCI. Takođe, isti WSFC klaster može pokrenuti više FCI (više grupa resursa), zavisno od hardverskog kapaciteta, kao što su CPU, memorija i broj diskova.

Binarni podaci proizvoda instalirani su lokalno na svakom čvoru FCI, a proces je sličan *standalone* instalacijama za SQL Servera. Međutim, tokom pokretanja, usluge se ne pokreću automatski, već njima upravlja WSFC.

FCI mora da koristi zajedničko skladište između svih čvorova za čuvanje baze podataka i logova. Deljeno skladištenje može biti u obliku WSFC klaster diskova, diskova na SAN-u (Storage Area Network), Storage Spaces Direct (S2D) ili deljenje datoteke na SMB-u. Na ovaj način, svi čvorovi u FCI imaju isti pogled podataka kad god dođe do prebacivanja. To,

međutim, znači da deljena memorija može biti jedinstvena tačka pada, a FCI zavisi od osnovnog rešenja za skladištenje da bi se osigurala zaštita podataka.

VNN (Virtual Networ Name) za FCI pruža jedinstvenu tačku za vezu za FCI. To omogućava aplikacijama da se povežu na VNN bez potrebe da znaju trenutno aktivni čvor. Kada dođe do greške, VNN se registruje na novi aktivni čvor nakon što je pokrenut. Ovaj postupak je transparentan za klijenta ili aplikaciju koja se povezuje na SQL Server i to umanjuje zastoj aplikacije ili klijenta tokom pada.

Šta SQL Server failover klaster obezbeđuje?

SQL Server failover klaster je takođe poznat kao klaster visoke dostupnosti, jer pruža redundantnost za kritične sisteme. Glavni koncept koji se nalazi iza klastera je uklanjanje jedinstvene tačke pada (eng. single point of failure) uključivanjem više mrežnih veza i zajedničkog prostora za čuvanje podataka spojenih preko SAN-a (Storage area network) ili NAS-a (Network attach storage).

Svaki čvor u okruženju klastera prati se sve vreme putem privatne mrežne veze zvane *heartbeat*. Sistem mora biti u stanju da prebrodi situaciju koja se zove *split-brain* koja se dogodi ako sve veze *heartbeat-a* padnu istovremeno. Onda svi drugi čvorovi mogu zaključiti da je jedan čvor pao i pokušati ponovo da pokrenu aplikaciju, svako ponaosob. Klaster koristi quorum za nadgledanje ukupnog zdravlja klastera i maksimizovanje tolerancije na greške na nivou čvora. Podešavanje quoruma je važno da ne bi više skupova čvorova moglo samostalno da uspostavi quorum. O quorumu će u kasnijem tekstu biti više reči.

Neka klaster CLUSTER-01 sadrži dva servera – čvora, Server1 i Server2, kao na slici 4. Postoji jedna SQL Server instanca SQL-INST-01. Takođe, postoji jedno deljeno skladište povezano sa svim serverima.

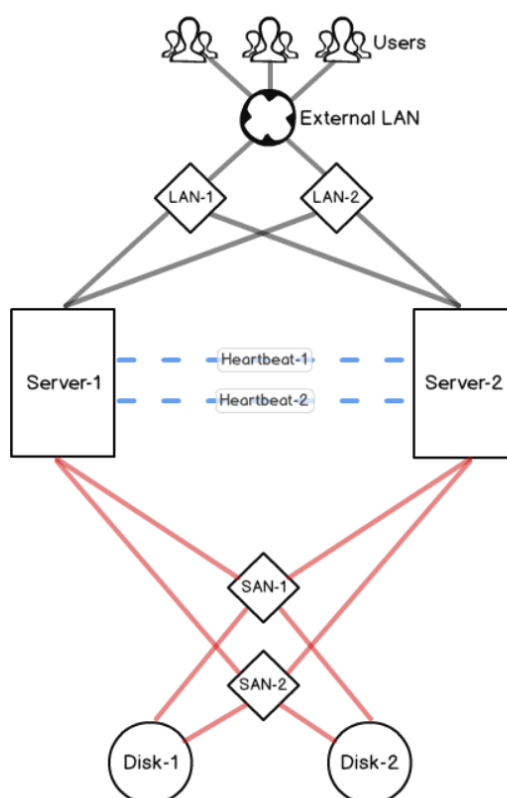
Pretpostavimo da se jedna instanca SQL Server pokreće na aktivnom čvoru klastera i da je pasivni čvor dostupan za preuzimanje kada je to potrebno. U ovom trenutku, aktivni čvor komunicira i sa bazom podataka i sa quorumom na deljenom memorijskom prostoru. Budući da samo jedan čvor istovremeno može pristupiti zajedničkoj deljenoj memoriji, pasivni čvor ne pristupa bazi podataka ili quorumu. Pored toga, aktivni čvor šalje *heartbeat* signale preko privatne mreže, a pasivni čvor ih nadgleda, tako da može preuzeti ako dođe do kvara. Klijenti takođe komuniciraju s aktivnim čvorom preko jedinstvenog imena SQL Servera i IP adrese tokom izvođenja neophodnih radnji.

Kada Server1 neočekivano padne, klaster usluge u CLUSTER-01 je svestan toga zbog *heartbeat-a*. Posle određenog vremena, pasivni čvor pretpostavlja da je aktivni čvor pao, inicira prelazak i automatski pokreće SQL Server instancu SQL-INST-01 na serveru Server2.

SQL-INST-01 instanca je pokrenuta i povezala se sa svim istim bazama podataka u zajedničkoj memoriji - postoji jedna kopija podataka koja se ne pomera. Kao deo procesa prelaska, pasivni čvor (sada aktivni čvor) preuzima kontrolu nad deljenom memorijom i čita quorum tražeći bilo kakve nesinhronizovane promene konfiguracije. Takođe preuzima

kontrolu nad klasteriranim imenom SQL Servera i IP adresom. Pored toga, kako čvor preuzima baze podataka, mora izvršiti pokretanje SQL Server-a i oporaviti baze podataka. Kao deo pokretanja SQL Servera, sve transakcije koje su bile u izvršenju i nisu završene u trenutku pada, vraćaju su unazad.

Dok se događalo ovo automatsko prebacivanje, korisnici nisu mogli da se povežu sa instancom SQL-INST-01. Međutim, nakon što je postala ponovo dostupna, mogli su normalno da nastave sa radom i nisu imali svest o tome da je server još uvek nedostupan.



Slika 4 – Klaster sa dva čvora

U SQL Server failover klasteru podaci moraju biti u deljenoj memoriji. Klaster može premestiti instancu SQL Servera, ako jedan čvor ima problema, jer se svi podaci dele. Ovo rešenje može garantovati duže vreme trajanja i redundantnost. Budući da postoji samo jedan prostor za skladištenje, potrebni su redovni zahtevi za održavanje SQL Servera. Takođe, ako deljena memorija nije redundantna, nakon pada skladišta, baza podataka SQL Servera neće biti dostupna. Za zahtevna okruženja SQL Servera, gde se prekid meri u sekundama, treba razmotriti vreme prelaska, jer promena između čvorova nije trenutna.

Klaster u osnovi daje mogućnost da svi podaci instance SQL Servera budu instalirani u nečemu što predstavlja deljeni deo kome se može pristupiti sa različitih servera. Uvek će imati isto ime instance, SQL Agent posla, povezane servere i loginove gde god bude

premešten. Može se, čak, učiniti da uvek koristi istu IP adresu i port, tako da nijedan korisnik SQL Servera ne mora da zna gde se u bilo kom trenutku nalazi.

Zašto je SQL klasterovanje korisno?

Klasterovanje je dizajnirano da poboljša dostupnost fizičkog hardvera servera, operativnog sistema i SQL Server instanci, ali ne i zajedničku memoriju. Ako bilo koji od ovih aspekata padne usled nekog problema, instanca SQL Servera takođe pada. Drugi čvor u klasteru automatski preuzima neuspelu instancu SQL Servera kako bi smanjio vreme zastoja na minimum. Slede prednosti klasterovanja:

- Kvarovi hardvera su noćna mora na samostalnim serverima. Ako server počne da ima probleme u klasteru, može se lako pokrenuti instanca SQL Servera iz drugog čvora dok se ne reši problem u čvoru.
- Ažuriranje dodatnih sigurnosnih delova na samostalnom serveru može biti vrlo naporno i neugodno za posao: SQL Server je nedostupan dok se čeka da se server ponovo pokrene. Korišćenjem klasterovanja, mogu se ažurirati sigurnosni delovi sa samo kratkim zastojima za aplikaciju, dok se instancu SQL Servera premešta u drugi čvor.
- Klasteri mogu takođe pružiti dodatni alat alatu za rešavanje problema. Na primer, ako se uoče visoke latencije prilikom korišćenja skladišta i isključeni su svi neposredni kandidati, može se preći na drugi čvor kako bi se uvidelo da li je problem sa komponentom na čvoru.
- Klasterovanje je transparentno za aplikaciju koja poziva neku uslugu. Mnogo stvari sa SQL serverom jednostavno funkcionišu s klasterovanjem, dok malo teže funkcionišu s drugim alternativama. Sa klasterovanjem, sve baze podataka, login podaci, agent poslovi i sve ostalo što je u instanci SQL Servera se prebacuje i sastaju se kao jedinstvena jedinica, odnosno nema potrebe da se išta od toga dodatno konfiguriše. Takođe se može klasterovati koordinator distribuiranih transakcija i da se i on prebaci s instancom.

Kao što je spomenuto klasterovanje je korisno kod ažuriranja sigurnosnih delova, a može pomoći da se smanji vreme zastoja kada se vrši bilo kakvo održavanje na čvorovima klastera. Na primer, ako treba da se ažurira ili nadogradi hardver na fizičkom serveru ili da se instaliraju novi servisni paketi u operativnom sistemu, to se može uraditi po principu jedan po jedan čvor:

1. Prvo se ažurira pasivni čvor koji na kome instanca SQL Servera nije aktivna.
2. Zatim se ručno prelazi iz aktivnog čvora u sada ažurirani čvor, koji postaje aktivni čvor.
3. Zatim se ažurira trenutno pasivni čvor.
4. Nakon što što je ažuriran, ako ima potrebe, korisnik može da vrati resurse na prvobitni čvor. Ova karakteristika klastera pomaže u smanjenju ukupnog zastoja prouzrokovano ažuriranjem.

Prilikom pokretanja nadogradnje treba osigurati da se ne prebaci na čvor koji nije nadograđen, jer bi to izazvalo nestabilnost.

Šta klasterovanje ne obuhvata?

Klasterovanje neće poboljšati performanse, osim ako se ne pređe na snažnije servere ili brži prostor za skladištenje istovremeno kada se primeni klasterovanje. Ako se koristila lokalna memorija, prelazak na SAN ne podrazumeva odlične performanse. Takođe, klasterovanje ne garantuje da je sve što je uključeno u SAN redundantno. Ako skladišni prostor postane nedostupan, i baza podataka će postati nedostupna. Klasterovanje nije zamišljeno da bi zaštitilo podatke, jer zajednička memorija predstavlja jedinstvenu tačku pada u klasterovanju. Podaci moraju biti zaštićeni pomoću drugih opcija, kao što su rezervne kopije, šipovanje logova ili disk mirroring. Zapravo, isti diskovi baze podataka se dele, iako ih istovremeno ne vide svi serveri u klasteru, tako da se korupcija u jednom disku prenosi na druge servere.

Klasterovanje ne štedi prostor ili napor za pravljenje rezervnih kopija ili za održavanje. I dalje je potrebno da se vrši redovno održavanje. Klasterovanje takođe neće pomoći u skaliranju čitanja. Iako se instanca SQL Servera može pokrenuti na bilo kojem čvoru klastera, instanca se pokreće samo na jednom čvoru. To skladište ne može čitati niko drugi u klasteru.

Na kraju, klasteri neće dati 100% poboljšanje vremena. Klasterovanje može ublažiti zastoje, ali ga ne može eliminisati. Postoje periodi zastoja kada se instanca SQL Servera prebacuje između čvorova. Na primer, sam prelazak uzrokuje prekid koji traje od nekoliko sekundi do nekoliko minuta, dok se usluge SQL Servera zaustavljaju na jednom čvoru, a zatim pokreću na drugom čvoru i vrši se oporavak baze podataka.

SQL Server failover klaster konfiguracije

Klasteri SQL Servera se često nazivaju ili Active/Active ili Active/Passive. U pravom smislu, to jednostavno znači da su svi čvorovi (serveri) koji učestvuju u klasteru SQL Servera posvećeni pokretanju najmanje jedne instance SQL Servera (Active-Active) ili je bar jedan od tih čvorova rezervisan kao *standby* da bi prihvatio prebacivanje SQL instance ako se dogodi.

Prednost Active/Active klastera je što se bolje iskoristi raspoloživi hardver. Oba čvora klastera se koriste umesto samo jednog, kao u Active/Passive klasteru. Nedostatak je što kada dođe do greške, obe instance SQL Servera rade na jednom fizičkom serveru, što može umanjiti performanse obe instance u kojima će možda biti potrebno podesiti memoriju kako bi se osiguralo da svaka ima odgovarajuću memoriju.

Postoje četiri glavne konfiguracije čvorova dostupne u failover klasteru u SQL Serveru: Active / Active (*multi-instance* klaster), Active / Passive, N + 1 i N + M.

Active / Active klaster, ili *multi-instance* klaster, deli resurse između virtuelnih servera. Svaki čvor istovremeno može hostovati dva ili više virtuelnih servera. Saobraćaj se može prebaciti na drugi aktivni čvor ili se može uravnotežiti preko preostalih čvorova ako postoji više aktivnih više čvorova.

Active / Passive klasteri za imaju rezervne čvorove koji se aktiviraju samo kada je primarni čvor nedostupan. Primarni čvor poseduje sve resurse. U slučaju kvara, čvor pripravnosti preuzima sve resurse i oporavlja bazu podataka iz datoteka baze podataka i logova transakcija.

N + 1 klaster je zasnovan na Active / Passive čvorovima gde dva ili više čvorova dele isti prelazni čvor. U situaciji kada svih N čvorova postane nedostupno, čvor u stanju pripravnosti mora biti sposoban da preuzme sav teret. Ovo se odnosi na klastere sa više instanci, jer u suprotnom ova konfiguracija postaje Active / Passive.

N + M klaster ima dva ili više aktivnih čvorova i dva ili više čvorova u stanju pripravnosti. Jeftiniji je za implementaciju od konfiguracije N + 1, jer se opterećenje može rasporediti na više čvorova u stanju pripravnosti.

Kvorum

Kvorum je osmišljen tako da spreči *split-brain* scenarije koji se mogu dogoditi kada postoji particionisanje u mreži i podskupovi čvorova ne mogu međusobno komunicirati. To može dovesti do toga da oba podskupa čvorova pokušaju da poseduju radno opterećenje i pišu na isti disk, što može dovesti do brojnih problema. Međutim, ovo je sprečeno konceptom kvoruma failover klastera koji prisiljava samo jednu od tih grupa čvorova da nastavi sa radom, tako da će samo jedna od tih grupa ostati na mreži.

Kad su oba čvora klastera pokrenuta i rade i učestvuju u svojim aktivnim i pasivnim ulogama, oni međusobno komuniciraju putem mreže. Na primer, ako se promeni konfiguracijska postavka na aktivnom čvoru, ta se konfiguracija automatski i brzo širi na pasivni čvor, osiguravajući na taj način sinhronizaciju.

Međutim, može se izvršiti promena na aktivnom čvoru i on može pasti pre nego što se promena pošalje putem mreže i izvrši na pasivnom čvoru. U ovom scenariju, promena se nikada ne primenjuje na pasivni čvor. U zavisnosti od prirode promene, to može prouzrokovati probleme, čak može dovesti do toga da i oba čvora klastera padnu.

Kvorum je u osnovi log datoteka, po konceptu slična logovima baze podataka. Njegova svrha je snimanje bilo kakvih promena izvršenih na aktivnom čvoru. Na ovaj način, ako bilo koja zabeležena promena ne dođe do pasivnog čvora, jer je aktivni čvor pao i ne može poslati promenu na pasivni čvor preko mreže, pasivni čvor, kada postane aktivni čvor, može pročitati log datoteku kvoruma da sazna šta je promena. Pasivni čvor tada može izvršiti promenu pre nego što postane novi aktivni čvor. Ako je stanje ovog pogona ugroženo, klaster može postati neoperativan.

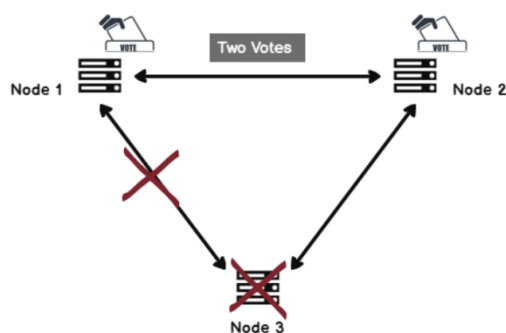
Zapravo, svaki kvorum (server) klastera može dati jedan glas, pri čemu većina ukupnog broja glasova (na osnovu broja ovih kvoruma klastera koji su na mreži) određuje da li klaster i dalje radi na čvoru klastera. Ovo sprečava da više čvorova klastera pokušava preuzeti vlasništvo nad istom instancom SQL Servera. Kvorumima za glasanje su čvorovi klastera ili, u nekim slučajevima, disk svedok ili deljena datoteka svedok. Svaki kvorum klastera za glasanje (s

izuzetkom deljene datoteke svedoka) sadrži kopiju konfiguracije klastera. Usluga klastera radi na tome da sve kopije budu sinhronizovane u svakom trenutku.

Slede četiri podržana načina kvoruma Windows failover klastera.

Node Majority

Svaki čvor koji je dostupan i u komunikaciji može glasati. Klaster funkcioniše samo sa većinom glasova. Ova konfiguracija kvoruma podržava gubitak polovine čvorova klastera nakon zaokruživanja na manji broj. Ovo je preporučena metoda za klastere sa neparnim brojem čvorova.

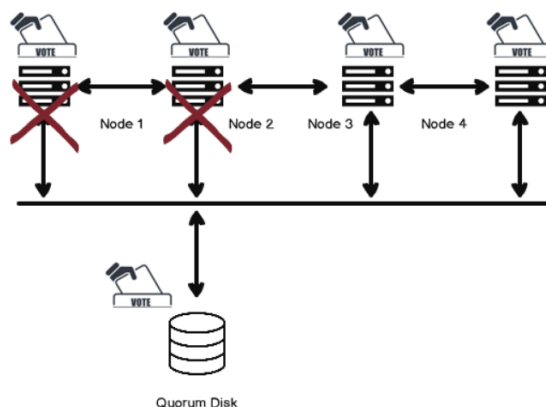


Slika 5 – Node Majority režim

Node and Disk Majority

Svaki čvor plus određeni disk u skladištu klastera (svedok disk) mogu glasati, kad god su dostupni i u komunikaciji. Ovaj disk može biti disk male veličine i velike dostupnosti i trebalo bi da bude deo klaster grupe. Pogodan je za klaster sa parnim brojem čvorova. Kvorum disk takođe čuva podatke o konfiguraciji klastera. Klaster funkcioniše samo sa većinom glasova.

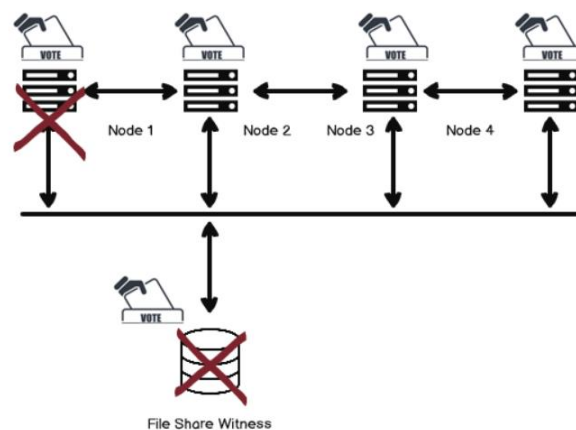
Kod ovog moda, klaster može da pretrpi otkaze polovine čvorova (zaokruživanjem naviše) ako disk ostane na mreži. Takođe, može da pretrpi otkaze polovine čvorova (zaokruživanjem naniže) ako disk svedok padne ili je nedostupan.



Slika 6 – Node and disk Majority režim

Node and File Share Majority

Svaki čvor plus imenovani zajednički fajl koji je kreirao administrator (deljena datoteka svedok) mogu glasati, kad god su dostupni i u komunikaciji. Klaster funkcioniše samo većinom glasova. Mogo bi postojati klaster u različitim regionima i možda ne bi bio deljen skladišni prostor između njih. Deljena datoteka svedok je deljena datoteka i može se konfigurirati na bilo kojem serveru koji je prisutan u aktivnom direktorijumu, a svi čvorovi klastera treba da imaju pristup toj deljenoj datoteci. Ova datoteka ne sadrži nikakve konfiguracione podatke. Aktivni čvor klastera stavlja zaključavanje na zajedničku datoteku, a deljena datoteka svedok sadrži podatke o vlasniku.

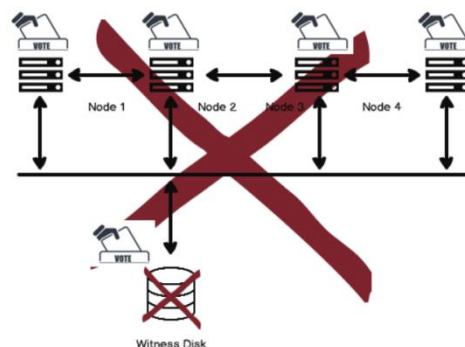


Slika 7 – Node and File Share Majority režim

No Majority, Disk Only

U ovoj konfiguraciji samo svedok disk ima glas, a čvorovi klastera ne mogu da deluju kao glasači. U ovom slučaju, svedok disk deluje kao jedinstvena tačka pada, a ako nije dostupan ili je oštećen, ceo klaster je oboren. Ovo nije preporučena konfiguracija kvoruma.

Kod klastera sa dva čvora, ova konfiguracija je najčešće korišćena konfiguracija kvoruma, obično poznata kao kvorum disk. Konfiguracija kvoruma može se promeniti nakon razmeštanja klastera na osnovu broja klasterovanih čvorova i korisničkih zahteva. Tako se u klasterima sa većim brojem čvorova od dva, druga tri kvorum režima češće koriste.



Slika 8 – No Majority režim

Dynamic Quorum configuration mode

U ovoj konfiguraciji, klaster dinamički izračunava kvorum prema stanju aktivnih glasača. Ako je čvor pao, grupa klastera ne broji njegov glas za proračun kvoruma. Jednom kada se čvor pridruži klasteru, dodeljuje mu se kvorumski glas. Mogu se povećati ili smanjiti glasovi kvoruma koristeći dinamički kvorum da bi se zadržao klaster. Glasovi kvoruma se dinamički podešavaju ako je zbir glasova manji od tri u konfiguraciji klastera sa više čvorova. Ovaj koncept ukupnog broja glasova koji se prilagođava nakon više uzastopnih neuspeha se zato naziva dinamički kvorum.

Korišćenjem dinamičkog svedoka, glasanje svedoka se dinamički prilagođava. Ako imamo neparan broj glasova kvoruma u konfiguraciji klastera, on ne koristi dinamičkog svedoka kvoruma. U slučaju parnog broja glasova, dinamički kvorum takođe daje glas. Može se konfigurisati ili deljena datoteka svedok ili disk svedok kao deo dinamičke konfiguracije svedoka. Takođe, može se koristiti *Cloud* svedok počevši od Windows Servera 2016.

Ovaj režim se zasniva na sledećim konceptima:

- Za parni broj čvorova, svaki čvor ima glas i dinamički glas svedoka
- Za neparan broj čvorova, svaki čvor ima glas, ali dinamičko glasanje svedoka se ne računa. (dinamički svedok glas = 0)

Dinamički kvorum omogućava dinamičko dodjeljivanje glasa čvoru kako bi se izbeglo gubljenje većine glasova i omogućilo da se klaster pokreće sa jednim čvorom (poznat kao last-man standing). Dinamički kvorum sarađuje sa dinamičkim svedokom na način opisan u nastavku:

- Ako postoji parni broj čvorova i nema svedoka, jednom čvoru se poništava glas. Na primer, samo tri od četiri čvora dobiju glasove, pa je ukupan broj glasova tri, a dva preživela čvora sa glasovima smatraju se većinom.
- Ako postoji neparan broj čvorova i nema svedoka, svi dobijaju glasove.
- Ako postoji parni broj čvorova plus svedok, svedok glasa, pa postoji neparan broj.
- Ako postoji neparan broj čvorova plus svedok, svedok ne glasa.

Ovi modovi i ručna dodela glasova se može podešavati ili putem klaster menadžmenta ili preko konzole. U nastavku su dati neki primeri podešavanja parametara.

- Postavljanje moda Node Majority:
Set-ClusterQuorum –Cluster CONTOSO-FC1 –NodeMajority
- Uklanjanje mogućnosti glasanja čvoru:
(Get-ClusterNode ContosoFCNode1).NodeWeight=0
- Omogućavanje dinamičkog kvoruma:
(Get-Cluster CONTOSO-FC1).DynamicQuorum=1

Ponovno pokretanje klastera

Budući da određeni klaster ima specifičan skup čvorova i određenu konfiguraciju kvoruma, softver klastera na svakom čvoru čuva informacije o tome koliko glasova čini kvorum za taj klaster. Ako se broj spusti ispod većine, klaster prestaje pružati usluge. Čvorovi će i dalje slušati dolazne veze s drugih čvorova na određenom portu, u slučaju da se ponovo pojave na mreži, ali čvorovi neće početi funkcionisati kao klaster dok se ne postigne kvorum.

Postoji nekoliko faza kroz koje klaster mora proći da bi postigao kvorum:

1. Kada se pojavi neki čvor, on određuje postoje li drugi članovi klastera sa kojima može komunicirati (ovaj proces može biti u toku na više čvorova istovremeno).
2. Jednom kada se uspostavi komunikacija sa drugim članovima, članovi upoređuju poglede svog članstva u klasteru sve dok se ne dogovore o jednom pogledu (na osnovu vremenskih žigova i drugih informacija).
3. Donosi se odluka da li ovaj skup članova ima kvorum ili, drugim rečima, ima dovoljno članova da *split-brain* scenario ne može postojati (skup čvorova ne može da komunicira s drugim skupom).
4. Ako nema dovoljno glasova za postizanje kvoruma, glasači čekaju da se pojavi više članova. Ako je prisutno dovoljno glasova, usluga klastera počinje da podiže klasterske resurse i aplikacije za pružanje usluga.
5. Sa postignutim kvorumom, klaster postaje potpuno funkcionalan

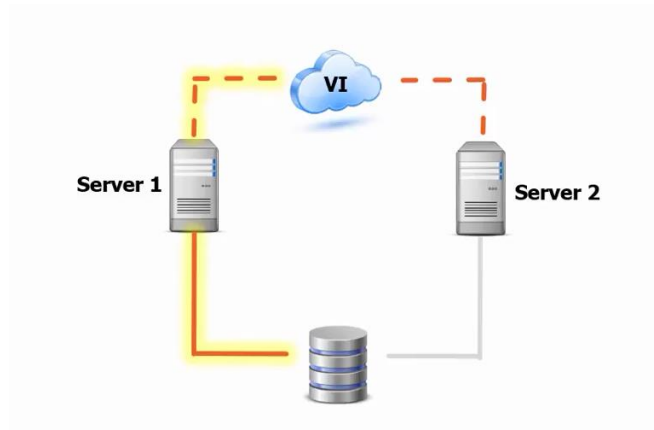
Klaster koji nema dovoljno glasova kvoruma se neće pokrenuti. Nakon što korisnik utvrdi da ne možete oporaviti klaster dovodeći čvorove ili kvorum svedoka u zdravo stanje, forsiranje startovanja klastera postaje neophodno. Prisiljavanje klastera da poništava postavke konfiguracije kvoruma klastera i pokreće klaster obavlja se u režimu *ForceQuorum*. Nakon što je korisnik prisilno pokrenuo klaster na čvoru, potrebno je pokrenuti sve preostale čvorove u klasteru sa postavkom za sprečavanje kvoruma. Čvor započet sa postavkom koja sprečava kvorum pokazuje usluzi klastera da se pridruži postojećem pokrenutom klasteru umesto da formira novu instancu klastera. To sprečava da preostali čvorovi formiraju razdvojeni klaster koji sadrži dve konkurentne instance.

Za sve kvorum modela sem *Disk Only*, efektivnost zavisi od komunikacije između čvorova koji glasaju. Loša veza može dovesti do toga da se neki čvor u drugoj podmreži može videti kao neaktivan iako to nije. Zavisno od topologije klastera i kvorum moda, može se desiti da bude kreirano više podmreža čvorova koji glasaju. Ovaj *split-brain* scenario moguć je samo kada administrator sistema ručno izvrši operaciju prisilnog kvoruma ili, u vrlo retkim okolnostima, kod prisilnog prebacivanja, eksplicitnom podelom skupa čvorova kvoruma.

Da bi se pojednostavila konfiguracija kvoruma i povećalo vreme trajanja, možda će biti korisno da korisnik podesi *NodeWeight* svakog čvora tako da se glas čvora ne ubraja u kvorum. Podešavanje se može vršiti uklanjanjem mogućnosti za glasanje nekom čvoru ili davanjem dodatnog glasa nekom čvoru.

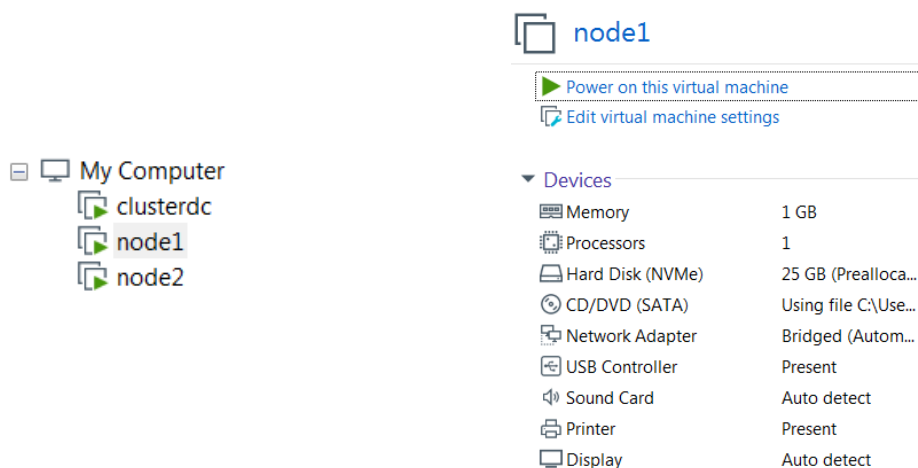
Kreiranje klastera

Pre nego što SQL Server failover klaster bude instaliran, neophodno je izabrati odgovarajući hardver i operativni sistem na kom će raditi SQL Server. Takođe, potrebno je konfigurisati klaster i podesiti mrežu. Za klaster je potrebna, pored čvorova i jedna virtualna instanca koja redirektuje saobraćaj ka aktivnom serveru (slika 9). Klijent komunicira sa klasterom preko ove instance, koja rutira zahtev ka aktivnom čvoru.



Slika 9 – Failover klaster sa dva čvora

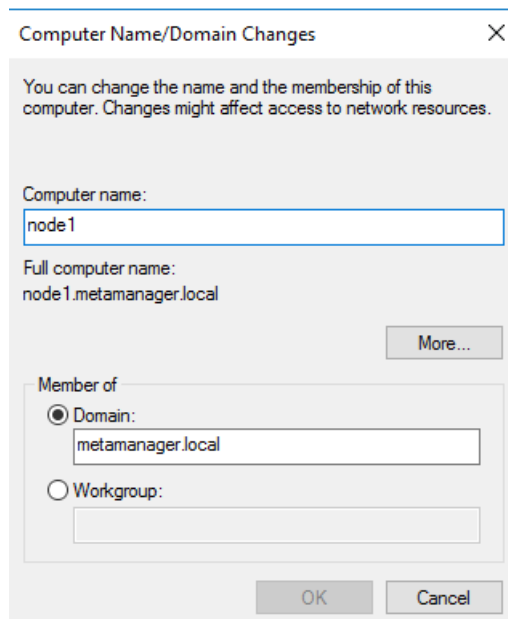
Kako bi bilo simulirano okruženje instaliran je *VMware workstation*. Pomoću ovog softvera su kreirana 2 virtualna čvora i na svakom od njih je instaliran operativni sistem Windows Server 2016. Pored toga, kreiran je i jedan server koji predstavlja kontroler domena (slika 10).



Slika 10 – Kreirane virtualne mašine

Za čvorove je kreiran još po jedan mrežni adapter koji se koristi za međusobnu, privatnu komunikaciju čvorova, odnosno za *heartbeat*. Virtualne mašine su podešene i dodeljene su im IP adrese i za svaki od čvorova je instalirana opcija za *failover* klaster.

Nakon toga su čvorovi pridruženi kontroleru domena (slika 11).



Slika 11 – Pridruživanje čvora 1 kontroleru domena

Zatim je potrebno kreirati deljenu memoriju i za tu potrebu je korišćen iSCSI (Internet Small Computer Systems Interface) target. Na kontroleru domena je instaliran iSCSI target server koji predstavlja udaljeno deljeno skladište. Nakon toga su kreirani iSCSI diskovi na targetu. Čvorovi se nazivaju iSCSI inicijatori, jer koriste diskove sa target servera. Kako bi čvorovi mogli da koriste diskove, oni su specificirani kao inicijatori na targetu i u svakom čvoru je specificiran target. Nakon formatiranja diskova kreiran je Windows klaster. Na slici 12 se može videti da klaster čine 2 čvora. Na slici 13 je su prikazani prethodno kreirani diskovi koji se koriste u klasteru. Takođe, na slici 14 se mogu videti 2 mreže, jedna javna i jedna privatna.

Name	Status	Assigned Vote	Current Vote	Site
node1	Up	1	1	
node2	Up	1	1	

Slika 12 - Čvorovi


Name	Status	Assigned To	Owner Node
Cluster Disk 1	Online	Disk Witness in Quorum	node1
Cluster Disk 2	Online	SQL Server (MSSQLSERV...	node1
Cluster Disk 3	Online	Available Storage	node1
Cluster Disk 4	Online	Available Storage	node1
Cluster Disk 5	Online	Available Storage	node1
Cluster Disk 6	Online	SQL Server (MSSQLSERV...	node1

Slika 13 – Diskovi

Name ▲	Status	Cluster Use	Information
Cluster Network 1	Up	Cluster and Client	
Cluster Network 2	Up	Cluster Only	

Slika 14 - Mreže

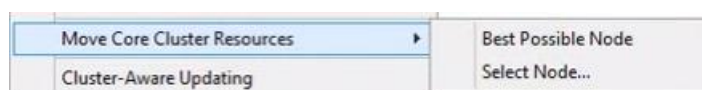
Na slici 15 se može videti da je čvor 2 trenutni aktivni čvor. Što se tiče kvoruma, korišćena su podrazumevana podešavanja, gde je disk 1 izabran kao disk svedok (node and disk majority). Disk svedok se može menjati ili se korisnik može izabrati deljeni fajl za svedoka, a takođe se mogu izabrati i ostala podešavanja kao što su glasovi diskova.


Summary of Cluster DEVSQLCLU
DEVSQLCLU has 0 clustered roles and 2 nodes.

Name: DEVSQLCLU.metamanager.local	Networks: Cluster Network 1, Cluster Network 2
Current Host Server: node2	Subnets: 2 IPv4 and 0 IPv6
Recent Cluster Events: None in the last 2 hours	Storage Spaces Direct (S2D): Disabled
Witness: Cluster Disk 1	


Slika 15 – Podaci o klasteru

Host čvor se može menjati selektovanjem odgovarajućeg čvora ili izborom opcije za automatsko selektovanje najboljeg čvora (slika 16). Za potrebe primera je čvor 1 izabran kao host čvor. Takođe, čvorovi se mogu dodavati i brisati iz klastera.



Slika 16 – Prebacivanje resursa klastera

Kada je potrebno instalirati novi apdejt na čvorovima, pored manualnog instaliranja na svakom čvoru, postoji i opcija *Cluster aware updating*. Izborom opcije *Apply updates to this cluster*, prvo se apdejtuju pasivni čvorovi (slika 17). Zatim se vrši prebacivanje usluga sa aktivnog čvora na neki drugi pasivni čvor, kako bi se i na njemu instalirao apdejt. Mešutim, nakon instaliranja, ne vraća usluge prethodno aktivnom čvoru.


DEVSQLCLU - Cluster-Aware Updating

Connect to a failover cluster:
DEVSQLCLU

Cluster nodes:

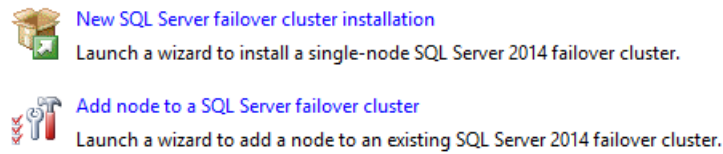
Node name	Last Run status	Last Run time
node1	Not Available	Not Available
node2	Not Available	Not Available

Cluster Actions

- Apply updates to this cluster
- Preview updates for this cluster
- Create or modify Updating Run Profile
- Generate report on past Updating Runs
- Configure cluster self-updating options
- Analyze cluster updating readiness

Slika 17 – Cluster Aware opcija

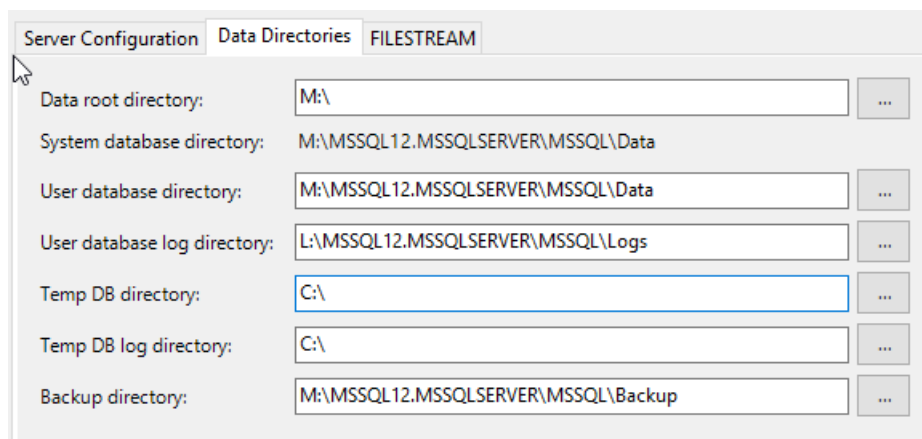
Nakon podešavanja klastera, potrebno je instalirati SQL Server na svim čvorovima. Na čvoru hostu je potrebno izabrati opciju za instaliranje single-node failover klastera i kada se tu izaberu odgovarajuća podešavanja i instalira SQL Server, na ostalim čvorovima je potrebno izabrati opciju dodavanja čvora postojećem klasteru (slika 18). Svaki novi čvor će pročitati konfiguraciju iz klastera, pa je instaliranje jednostavno.



Slika 18 – Instaliranje SQL Servera

Potrebno je navesti ime SQL Server mreže, što je zapravo ime klastera i to ime će koristiti klijentska aplikacija kako bi se konektovala na klaster. U ovom slučaju, ime je VSQL.

Potrebno je izabrati i diskove koji su prethodno kreirani, a na kojima će se pamtiti podaci baze podataka. Ranije su kreirani diskovi M i L namenjeni za podatke, odnosno logove, što je i izabrano (slika 19).



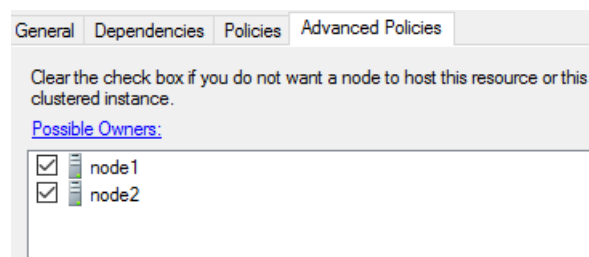
Slika 19 – Izbor diskova za čuvanje baze

Instalacijom SQL Servera na oba čvora, kreiran je SQL Server klaster. Za aktivni čvor se mogu videti dodeljeni resursi koji obuhvataju diskove, SQL Server ime, preko koga klijent pristupa i SQL Server usluge kojima upravlja klaster (slika 20) . Svi ovi resursi zajedno čine jednu grupu resursa.



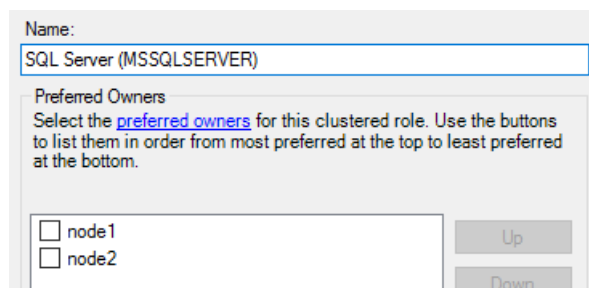
Slika 20 – Grupa resursa klastera

Grupa resursa se nalazi na čvoru 1, međutim moguće je prebaciti je na drugi čvor. Kako su oba čvora čvorovi klastera i čvor 2 može biti host (slika 21).



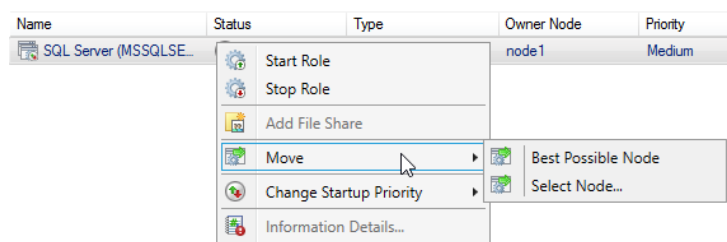
Slika 21 – Mogući vlasnici grupe resursa

Ukoliko je jedan čvor dosta bolji od drugih i korisnik želi da on bude host kada god je to moguće, može se postaviti čvor kao omiljeni čvor (slika 22). Tada će se izvršiti prebacivanje čim on postane dostupan.



Slika 22 – Omiljeni čvorovi

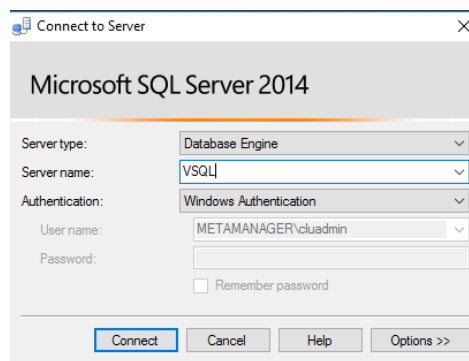
Izvršićemo prebacivanje SQL Server klastera sa čvora 1 na čvor 2 (slika 23). Tada prvo diskovi postaju dostupni na drugom čvoru, zatim ime mreže i na kraju SQL Server usluge.



Slika 23 – Prebacivanje grupe resursa

Važno je napomenuti da SQL Server klaster i Windows klaster ne moraju biti na istom čvoru. Sada kada se vrši prebacivanje resursa, potrebno je prebaciti celu grupu resursa, odnosno SQL Server ulogu, jer SQL Server zavisi od dodeljenih diskova, pošto na njima pamti podatke. S druge strane, ostali diskovi mogu biti na drugom čvoru, kao npr. kvorum disk koji se nalazi na čvoru koji je host Windows klastera.

Instalacijom SQL Servera, instaliran je i SQL Server menadžment, pa se pomoću njega može konektovati na klaster, navođenjem prethodno određenog imena – VSQL (slika 24).



Slika 24 – Povezivanje sa klasterom

Na slikama su prikazani primeri informacija o klasteru dobijenih preko SQL Server menadžmenta. U međuvremenu je izvršeno prebacivanje klastera sa čvora 1 na čvor 2 (slika 26), a to nije zahtevalo ponovnu konekciju korisnika, dakle, korisnik nije svestan klastera i normalno izvršava upite i dobija željene podatke iz baze.

`SELECT @@SERVERNAME AS ServerName`

	ServerName
1	VSQL

Slika 25 – Ime servera

`SELECT * FROM sys.dm_os_cluster_nodes`

	NodeName	status	status_description	is_current_owner
1	node1	0	up	0
2	node2	0	up	1

Slika 26 – Podaci o čvorovima

`SELECT * FROM sys.dm_io_cluster_shared_drives`

	DriveName
1	M
2	L
3	G
4	J
5	O
6	E

Slika 27 - Diskovi

Zaključak

U ovom radu je opisan mehanizam klastera kod SQL Servara. Na primerima je demonstrirano kreiranje klastera i simulirano je prebacivanje resursa sa jednog na drugi čvor. Zaključno, klasteri mogu doprineti dostupnosti resursa, jer u slučaju kvara servera ili tokom postupka održavanja, upravljanje se može preneti drugom serveru bez većih zastoja u pružanju usluga. Za vreme popravke ili održavanja servera koji je prethodno pružao usluge, klijent neće morati da čeka i nesmetano može da koristi usluge koje mu sada omogućava drugi server.

Literatura

[1] – Microsoft SQL Server oficijalna dokumentacija

<https://docs.microsoft.com/en-us/sql/sql-server/failover-clusters/windows/always-on-failover-cluster-instances-sql-server?view=sql-server-ver15#FCIelements>

[2] – Database Clustering Tutorial, Caleb Curry

<https://www.calebcurry.com/database-clustering-tutorial-1-intro-to-database-clustering/>

[3] – What is Clustering in DBMS, Quora

<https://www.quora.com/What-is-clustering-in-DBMS>

[4] – Windows failover cluster quorum modes, SQLShack

<https://www.sqlshack.com/windows-failover-cluster-quorum-modes-in-sql-server-always-on-availability-groups/>

[5] – What is a SQL Server failover clustering

<https://www.sqlshack.com/sql-server-failover-clustering/>

[6] – How SQL Server Cluster works, Logicalread

<https://logicalread.com/how-sql-server-clustering-works-part-1-w02/#.XvCMXpozZEZ>

[7] – Install SQL Server 2014- Install a cluster instance

<https://www.sqlshack.com/sql-server-2014-install-clustered-instance-step-step-13/>