



Univerzitet u Nišu
Elektronski fakultet
Katedra za računarstvo



SISTEMI ZA
UPRAVLJANJE
BAZAMA PODATAKA

Oracle DBMS
Cluster
rešenja

Profesor:

Prof.dr Aleksandar Stanimirović

Student:

Bojana Svilenković 1417

Niš, 2022.

SADRŽAJ

UVOD.....	3
1. DEFINICIJA PROBLEMA I FAZE.....	4
2. TIPOVI PODATAKA I NJIHOVE MERE.....	5
2.1. Klasifikacija na osnovu veličine domena	5
2.2. Kategorizacija klusterskih tehnika i prethodni rad	6
3. TEHNIKE KLASTERISANJA RUDARENJA PODATAKA	7
4. PREGLED ORACLE RAC-A.....	10
4.1. Pregled Oracle Clustervare-a za Oracle RAC	12
4.2. O Oracle RAC softverskim komponentama.....	13
4.3. O Oracle RAC pozadinskim procesima	13
4.4. Oracleova arhitektura paralelnog izvršavanja	14
4.4.1. Kako funkcioniše paralelno izvršavanje	15
4.4.2. Paralelne operacije i stepen paralelizma (DOP)	15
4.5. Paralelizam, Oracle particionisanje i Oracle RAC	16
4.5.1. Koristite particione spojeve	17
4.5.2. Koristite grupe paralelnih instanci (u Oracle bazi podataka 10g).....	17
4.5.3. Razmatranja o međusobnom povezivanju	18
4.6. Automatsko upravljanje opterećenjem	20
4.6.1. Usluge	20
ZAKLJUČAK	22
LITERATURA	23

UVOD

Pojam klastera se široko koristi u današnjoj ekonomiji, međutim samo mali procenat ljudi je svestan njegovog porekla i istorije. Reč "klaster" potiče iz engleskog jezika i direktno se prevodi kao "klaster" na srpski. Već možemo konstruisati sliku o tome kakav je klaster institucija zasnovan na ovim informacijama: grupisanje institucija zasnovano na komercijalnoj osnovi. U literaturi se često koristi izraz „aglomeracija firmi“, ali se odnosi isključivo na ekonomske definicije klastera kao koncepta i organizacionog oblika. Majkl E. Porter je skovao termin „poslovni klaster“ da bi opisao „prostornu koncentraciju međusobno povezanih preduzeća i srodnih institucija“. Prateći Porterovo vođstvo, brojni drugi naučnici su poboljšali fundamentalni koncept klastera zasnovan na aspektu i širini posmatranja. Klasteri se mogu grupisati oko istih, uporedivih ili čak različitih aktivnosti, što dovodi do toga da su današnji klasteri organizovana mešavina različitih aktivnosti povezanih intrigantnim vezama kao što su znanje, veštine, zahtevi, tržište i tako dalje. Nekoliko studija je pokazalo da udruživanje u klastere poboljšava kvalitet i uspeh aktivnosti oko kojih se klaster gradi. Klasteri se, s druge strane, definišu kao „agregacije brojnih organizacija i interesnih grupa koje su (regionalno) susedne jedna drugoj i imaju suštinske i strukturne veze jedna sa drugom jer njihove aktivnosti prate jedan ili više lanaca vrednosti. Klaster, prema ovoj definiciji, je grupa preduzeća koja aktivno učestvuju u lancu vrednosti. Aktivnosti članica klastera su međusobno komplementarne i integrisane u korist svih članica, što je važan atribut ovakvog klastera. Klasteri su, prema Evropskoj komisiji, „grupe nezavisnih kompanija – inovativnih start-upova, malih, srednje i velike kompanije i istraživačke organizacije - koje rade u određenom sektoru i regionu, a dizajnirane su da stimulišu inovativne aktivnosti promovisanjem intenzivnih interakcija, deljenja objekata i razmene znanja i stručnosti, i doprinoseći efikasnom transferu tehnologije, umrežavanju i širenje informacija među kompanijama u tom sektoru i regionu“. Iako se često pominju „virtuelni klasteri“ preduzeća koji pokrivaju veći geografski region, prostorna koncentracija je u srcu strategije klastera. Ovi klasteri su nastali kao rezultat napretka trenutnih informacionih i komunikacionih tehnologija (IKT), koje omogućavaju preduzećima da budu grupisana zajedno bez obzira na njihovu fizičku lokaciju.

1. DEFINICIJA PROBLEMA I FAZE

Grupisanje se može definisati na različite načine. Koristićemo primer koji su predstavili Jain i Dubes: „Klaster analiza organizuje podatke apstrahovanjem osnovne strukture, koja se može izraziti kao grupisanje osoba ili hijerarhija grupa.“ Reprezentacija se zatim može ispitati da se vidi da li se podaci grupišu u skladu sa prethodnim pojmovima ili da se predloži nova ispitivanja."

Analiza klastera, ukratko, deli stavke podataka u klastere, pri čemu su objekti koji pripadaju istom klasteru slični, a oni koji pripadaju drugim klasterima su različiti. U narednom odeljku biće razjašnjeni koncepti sličnosti i različitosti. Grupisanje ne može biti proces u jednom koraku, prema gornjoj definiciji. Jain i Dubes, u jednom od ključnih radova o analizi klastera, podelili su proces grupisanja u sledeće faze:

- Prikupljanje podataka podrazumeva pažljivo izdvajanje relevantnih stavki podataka iz osnovnih izvora podataka.
- Objekti podataka se u našem kontekstu razlikuju po svojim jedinstvenim vrednostima za skup karakteristika (ili mera).

Početni skrining: Ovo je proces manipulacije podacima nakon što su izvučeni iz izvora. Ova faza je povezana sa procesom čišćenja podataka, koji se obično koristi u skladištenju podataka.

Reprezentacija: Ovo uključuje pripremu podataka tako da budu prikladni za metod grupisanja. Bira se mera sličnosti i istražuju se svojstva i dimenzionalnost podataka.

Sklonost grupisanju: Određuje da li podaci koji se razmatraju imaju prirodnu tendenciju da se grupišu. Ova faza se ponekad zanemaruje, posebno kada se radi o velikim skupovima podataka.

Strategija grupisanja: Ovo podrazumeva odabir tehnike grupisanja i početnih parametara sa razmatranjem.

Validacija: Ovo je poslednja i, po našem mišljenju, najmanje istražena faza. Ručna inspekcija i vizuelni pristupi se često koriste u validaciji. Međutim, kako se obim podataka i njihova složenost povećavaju, nećemo moći da uporedimo rezultate sa unapred stvorenim shvatanjima ili drugim grupama. Tumačenje: Ova faza podrazumeva kombinovanje podataka u grupisanju sa nalazima drugih studija, kao što je klasifikacija, kako bi se izveli zaključci i dale preporuke za dodatna istraživanja. Ponuđena rešenja za gore pomenute korake su predstavljena u odeljcima koji slede. Počinjemo prikupljanjem podataka i ispitivanjem njihovih vrsta i metrika, koji su navedeni u sledećem odeljku.

2. TIPOVI PODATAKA I NJIHOVE MERE

Predmeti analize u grupisanju mogu uključivati ljude, plate, mišljenja, softverske entitete i razne druge stvari. U pogledu kvaliteta, ovi artefakti moraju biti izlagani sa pažnjom. Ovi kvaliteti su glavne varijable problema, a način na koji su izabrani ima značajan uticaj na rezultate algoritma grupisanja. Temeljna klasifikacija mnogih tipova promenljivih uključenih u većinu skupova podataka je koristan alat za otkrivanje neslaganja između delova podataka. Veličina domena i skala merenja se koriste za kreiranje kategorizacije.

2.1. Klasifikacija na osnovu veličine domena

Ova klasifikacija kategorizuje stavke podataka prema veličini njihovog domena ili broju jedinstvenih vrednosti koje mogu da zauzmu. Pretpostavićemo bazu podataka D sa n objekata, ili torkova, za ostatak ovog članka. Ako su k , i i z tri D stavke podataka, svaka od njih ima oblik: $k = (k_1; k_2; \dots; k_k)$, $i = (i_1; i_2; \dots; i_k)$ i $z = (z_1; z_2; \dots; z_k)$, gde je k dimenzionalnost i svaki ki , ii i zi , $1 \leq k$, „Tipovi podataka“ će se od sada nazivati „tipovima podataka atributa“. Dostupne su sledeće klase:

1. Atribut je neprekidan ako je njegov domen neprebrojivo beskonačan, odnosno, njegovi elementi se ne mogu preslikati u skup pozitivnih celih brojeva jedan prema jedan. Ovo ukazuje da postoji beskonačan broj vrednosti između bilo koje dve vrednosti atributa. Temperatura, boja i jačina zvuka su primeri takvih karakteristika.
2. Atribut je diskretan ako je njegov domen konačan skup, odnosno skup čiji članovi mogu biti preslikani u konačni podskup pozitivnih celih brojeva u korespondenciji jedan-na-jedan. Broj dece u domaćinstvu ili serijski brojevi publikacija su dva primera. Binarni atributi su karakteristike sa domenom koji sadrži tačno dve diskretne vrednosti. Oni su podskup diskretnih karakteristika, a mi koristimo odgovore da/ne na anketu i unose muškog/ženskog pola u bazi podataka osoblja kao primere.

2.2. Kategorizacija klusterskih tehnika i prethodni rad

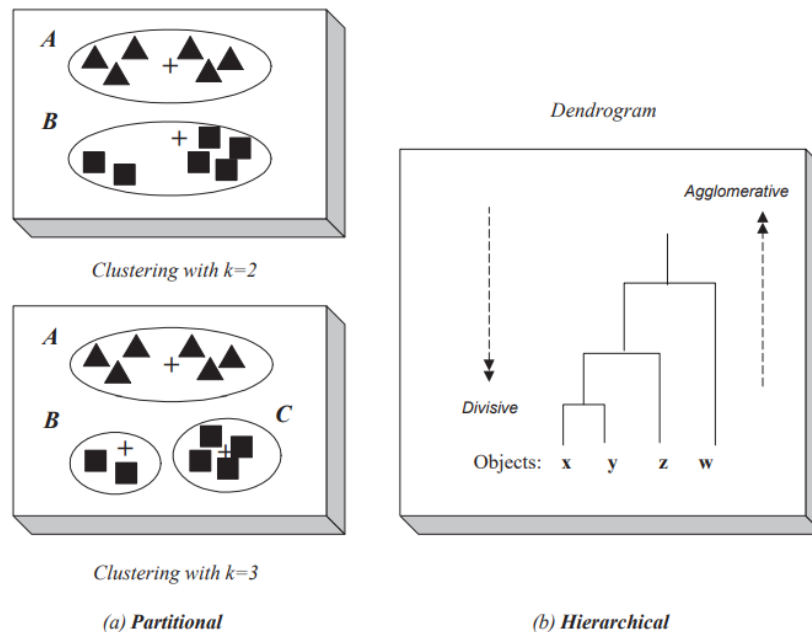
Da bi se pronašle koherentne grupe u ogromnim skupovima podataka, pojavile su se različite strategije. Dva tradicionalna pristupa grupisanja, kao i oni konkretniji, predstavljani su u naredna dva odeljka.

Osnovna tehnika grupisanja

Tehnike particionog i hijerarhijskog grupisanja su dve vrste pristupa grupisanja koje identifikujemo. Slede njihove definicije:

Razdvajanje: Metoda particionog klasterisanja kreira k podela baze podataka od n stavki, pri čemu svaki klaster optimizuje kriterijum grupisanja unutar svakog klastera, kao što je minimizacija zbira kvadrata udaljenosti od srednje vrednosti. Jedan od problema ovakvih algoritama je njihova ogromna složenost, jer neki od njih imaju za cilj da identifikuju globalni optimum potpunim nabranjanjem svih potencijalnih grupa. Broj podela je ogroman, čak i za skroman broj artikala. Kao rezultat toga, većina rešenja počinje grubom podelom, koja je generalno nasumična, a zatim je precizira.

- (a) Aglomerativni algoritmi počinju tako što tretiraju svaku stavku kao poseban klaster, a zatim kombinuju grupisanje na osnovu metrike udaljenosti. Grupisanje se može završiti kada se sve stavke nalaze u jednoj grupi ili u bilo koje vreme koje korisnik želi. Pohlepno spajanje odozdo prema gore koristi se u većini ovih strategija.
- (b) Divizionni algoritmi, s druge strane, koriste suprotan pristup. Počinju sa jednom grupom svih stvari i dele je na manje grupe dok svaki objekat ne padne u jednu grupu, ili po želji. U svakoj fazi, podela stavki podataka u različite grupe i praćenje istog obrasca dok svi objekti ne padnu u drugi klaster je strategija podele. Ovo je uporedivo sa strategijom zavadi pa vladaj koju koriste algoritmi zavadi pa vladaj. Obe tehnike su obično otežane činjenicom da kada se spajanje ili podela izvrši, to se ne može poništiti ili poboljšati. Slika 1(a) prikazuje primer dva algoritma za razdvajanje, svaki sa različitim početnim podešavanjima, koji rade na istom skupu podataka. Centar klastera je označen simbolom "+", koji je u ovom slučaju definisan kao srednja vrednost vrednosti klastera. Slika 1(b) prikazuje dendrogram generisan ili metodom razdvajanja ili aglomerativnog grupisanja u isto vreme.



Slika 1(a) i 1(b). Primeri klasičnih algoritama za grupisanje, gde je k broj klastera

Mogu se integrisati particione i hijerarhijske metode. To bi značilo da se rezultat dat hijerarhijskom metodom može poboljšati particionim korakom, koji prečišćava rezultat putem iterativnog premeštanja od bodova.

3. TEHNIKE KLASTERISANJA RUDARENJA PODATAKA

Osim dva osnovna tipa particionih i hijerarhijskih algoritama za grupisanje, pojavili su se još brojniji pristupi klaster analizama, od kojih se svaki fokusira na jedinstveno pitanje ili skup podataka je jedan od ovih pristupa.

Grupisanje na osnovu gustine: Ove metode grupišu objekte na osnovu ciljnih funkcija gustine. Broj stavki u određenoj blizini objekta podataka obično se karakteriše kao gustina. U ovim metodama, određeni klaster nastavlja da se razvija sve dok broj objekata u njegovoj blizini prelazi određeni prag. Ovo se razlikuje od pojma koji se krije iza particionih algoritama, koji koriste iterativno premeštanje tačaka datim određenim brojem klastera.

Grupisanje zasnovano na mreži: Ove metode se prvenstveno bave prostornim podacima, koji uključuju informacije o geometrijskoj strukturi objekata u prostoru, kao i njihove odnose, attribute i operacije. Cilj ovih algoritama je da podele skup podataka na određeni broj ćelija, a zatim operišu sa stavkama unutar tih ćelija. Oni ne pomeraju tačke; umesto toga, oni stvaraju brojne hijerarhijske nivoe grupisanja objekata. Na ovaj način, oni su slični hijerarhijskim

algoritmima, ali je spajanje mreža, a samim tim i klastera, određeno određenim parametrom, a ne merom udaljenosti. Grupisanje zasnovano na modelu: Ovi algoritmi traže odlične aproksimacije parametara modela koji najbolje odgovaraju podacima. Mogu biti particioni ili hijerarhijski, u zavisnosti od strukture ili modela koji predlažu za skup podataka i kako ga preciziraju da bi pronašli particionisanje. Oni su slični algoritmima zasnovanim na gustini po tome što generišu određene klastere kako bi poboljšali očekivani model. Oni, međutim, povremeno počinju sa određenim brojem klastera i možda neće koristiti istu ideju gustine.

Grupisanje kategoričkih podataka: Ove metode su kreirane za podatke u kojima se ne mogu koristiti euklidske ili druge numerički orijentisane metrike udaljenosti. Identifikujemo načine koji su slični i particionim i hijerarhijskim metodama u literaturi. Postoji nekoliko potkategorija za svaku kategoriju, kao što je grupisanje zasnovano na gustini usmereno ka geografskim podacima i metode za pronalaženje klastera. Tehnike kategoričkih podataka su izuzetak od ovog pravila. Pošto u takvim podacima ne postoji intrinzična geometrijska organizacija, tehnike koje su se pojavile u literaturi se uglavnom oslanjaju na pojmove koje prenose podaci, kao što su ko-pojavljivanja u torkama. S druge strane, skupovi kategorija podataka su brojni. Štaviše, određeni skupovi podataka, kao što je skup podataka popisa Sjedinjenih Država i skupovi podataka koji se koriste u otkrivanju šeme, imaju kombinaciju tipova atributa. Kao što će biti pomenuto, sadašnje metode grupisanja se koncentrišu na slučajeve kada su svojstva objekta sve iste vrste. Smatramo da je klaster analiza skupova podataka kategoričkog i mešovitog tipa fascinantna predmet u rudarenju podataka. Ali koji faktori doprinose efikasnosti i efektivnosti algoritma za grupisanje? Odgovor je dvosmislen. U zavisnosti od količine i dimenzionalnosti podataka, kao i funkcije cilja i korišćenih struktura, tehnika može dobro da funkcioniše na jednom skupu podataka, ali loše na drugom. Koji god pristup da se koristi, akademici se slažu oko sledećih karakteristika efikasne tehnike grupisanja:

- Skalabilnost se odnosi na kapacitet algoritma da rukuje velikim brojem stavki podataka (torke).
- Ispitati različite tipove atributa: mogućnost ispitivanja pojedinačnih tipova atributa kao i kombinacija tipova atributa.
- Pronaći klastere proizvoljnog oblika: oblik obično odgovara vrstama klastera koje algoritam može naći i ovo bi trebalo da smatramo veoma važnom stvari pri izboru metode, pošto želimo da bude što opštiji, različite vrste algoritama će biti pristrasne ka pronalaženju različitih vrsta klusterskih struktura/oblika i nije uvek lak zadatak odrediti oblik ili odgovarajuću pristrasnost. Pogotovo kada su prisutni kategorični atributi možda nećemo moći da razgovaramo o klaster strukturama.
- Minimalni zahtevi za ulazne parametre: Mnogi algoritmi za grupisanje zahtevaju neke korisnički definisane parametara, kao što je broj klastera, u cilju analize podataka. Međutim, sa velikim podacima skupova i veće dimenzionalnosti, poželjno je da metoda

zahteva samo ograničeno vođenje od strane korisnika, kako bi se izbegla pristrasnost u pogledu rezultata.

- Rukovanje šumom: Algoritmi za grupisanje treba da budu u stanju da obrađuju odstupanja, kako bi poboljšali kvalitet klastera. Odstupanja se definišu kao objekti podataka koji odstupaju od opšteprihvaćenih normi ponašanja i nazivaju se i izvanrednim. Detekcija odstupanja se smatra posebnim problemom.
- Osetljivost na redosled ulaznih zapisa: Isti skup podataka, kada se predstavi određenim algoritmima u različitim naložima, mogu proizvesti dramatično različite rezultate. Redosled unosa uglavnom utiče na algoritme koji zahtevaju jedno skeniranje skupa podataka, što dovodi do lokalno optimalnih rešenja na svakom koraku. Stoga je ključno da algoritmi budu neosetljivi na redosled unosa.
- Visoka dimenzionalnost podataka: Broj atributa/dimenzija u mnogim skupovima podataka je veliki, i mnogi algoritmi za grupisanje ne mogu da obrađuju više od malog broja (osam do deset) dimenzija. To predstavlja izazov grupirati skupove podataka visokih dimenzija, kao što je skup podataka popisa u SAD koji sadrži 138 atributa. Pojava velikog broja atributa se često naziva prokletstvom dimenzionalnosti. Ovo se koristi u sledećim prilikama:
 1. Kako broj atributa postaje sve veći, količina resursa potrebnih za njihovo skladištenje ili predstavljanje raste.
 2. Udaljenost date tačke od najbližeg i najdaljeg suseda je skoro ista, za a širok spektar distribucija i funkcija udaljenosti.Oba gorenavedena pojma veoma utiču na efikasnost algoritma za grupisanje, jer bi mu bilo potrebno više vremena za obradu podataka, dok bi u isto vreme nastali klasteri bili veoma lošeg kvaliteta.
- Interpretabilnost i upotrebljivost: U većini slučajeva očekuje se da algoritmi grupisanja daju upotrebljive i interpretabilne rezultate. Ali kada je u pitanju poređenje rezultata sa unapred stvorenim idejama ili ograničenjima, neke tehnike ne uspevaju da budu zadovoljavajuće. Stoga su lako razumljivi rezultati visoko poželjni.

Imajući u vidu navedene karakteristike, predstavljamo neke od najvažnijih algoritama koji imaju uticaja na klastersku zajednicu. Pokušaćemo da ih kritikujemo i saopštimo koji od uslova ispunjavaju ili ne ispunjavaju.

4. PREGLED ORACLE RAC-A

Klaster se sastoji od više međusobno povezanih računara ili servera koji krajnjim korisnicima i aplikacijama izgledaju kao da su jedan server. Oracle RAC¹ omogućava da se grupiše Oracle baza podataka. Oracle RAC koristi Oracle Clustervare za infrastrukturu za povezivanje više servera kako bi oni funkcionisali kao jedan sistem. Oracle Clustervare je prenosivo rešenje za upravljanje klasterima koje je integrisano sa Oracle Database. Oracle Clustervare je takođe neophodna komponenta za korišćenje Oracle RAC-a. Pored toga, Oracle Clustervare omogućava i neklasterskim Oracle bazama podataka i Oracle RAC bazama podataka da koriste Oracle infrastrukturu visoke dostupnosti. Oracle Clustervare omogućava da se kreira klasterizovani skup skladišta koji će koristiti bilo koja kombinacija neklasternih i Oracle RAC baza podataka.

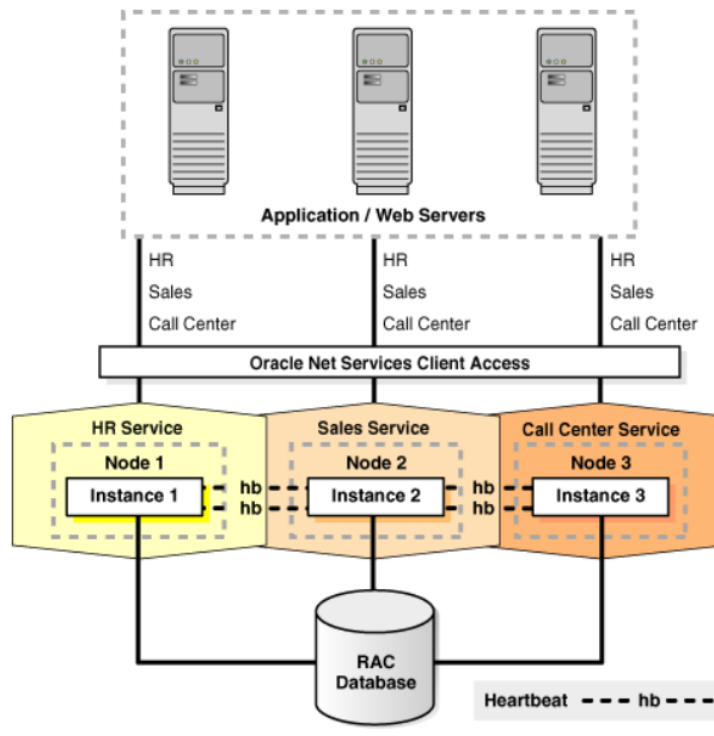
Oracle Clustervare je jedini klasterver koji je potreban za većinu platformi na kojima Oracle RAC radi. Takođe može se koristiti klasterver od drugih dobavljača ako je klasterver sertifikovan za Oracle RAC.

Neklasterske Oracle baze podataka imaju odnos jedan-na-jedan između Oracle baze podataka i instance. Oracle RAC okruženja, međutim, imaju odnos jedan-prema-više između baze podataka i instanci. Oracle RAC baza podataka može imati do 100 instanci, od kojih sve pristupaju jednoj bazi podataka. Sve instance baze podataka moraju da koriste istu međukonekciju, koju takođe može da koristi Oracle Clustervare.

Oracle RAC baze podataka se arhitektonski razlikuju od neklasterskih Oracle baza podataka po tome što svaka instanca Oracle RAC baze podataka takođe ima:

- Najmanje jedna dodatna nit ponavljanja za svaku instancu
- Tabelarni 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.

¹ Oracle RAC-Oracle Real Application Clusters



Slika 2. Oracle baza podataka sa Oracle RAC arhitekturom

Oracle RAC podešavanje obično zauzima jedan centar podataka. Ipak, možete da podesite Oracle RAC na klasteru sa proširenim rastojanjem, dizajnu koji nudi neverovatno brz oporavak od greške na lokaciji i omogućava svim čvorovima, na svim lokacijama, da aktivno izvršavaju transakcije kao deo jednog klastera baze podataka. Čvorovi klastera se nalaze u dve zgrade koje su udaljenije u proširenim klasterima (bilo gde preko ulice, preko kampusa ili preko grada). Podaci moraju biti prisutni na obe lokacije iz razloga dostupnosti, što zahteva upotrebu tehnologije preslikavanja diska za skladištenje. Visoku dostupnost i skalabilnost za sve vrste aplikacija obezbeđuje Oracle RAC. Da bi se primenila Oracle korporativna mrežna računarska arhitektura, Oracle RAC infrastruktura je takođe ključni element. Server je zaštićen od toga da postane jedinstvena tačka kvara dozvoljavajući brojnim instancama da pristupe jednoj bazi podataka. Sa Oracle RAC-om, mogu se razviti skalabilna okruženja koja podržavaju kritične poslovne aplikacije tako što će grupirati manje robne servere. Aplikacije koje instalirate na Oracle RAC baze podataka mogu da rade bez potrebe za modifikacijom koda.

4.1. Pregled Oracle Clustervare-a za Oracle RAC

Na svim sistemima Oracle Database, Oracle Clustervare nudi sveobuhvatno, integrisano rešenje za administraciju klastera. Sve mogućnosti neophodne za administriranje baze podataka klastera su obezbeđene ovom mogućnošću klastera, uključujući članstvo u čvorovima, grupne usluge, globalno upravljanje resursima i funkcije visoke dostupnosti.

Instalacija Oracle Clustervare-a je preduslov za instalaciju Oracle RAC-a, ali se može uraditi i odvojeno. Osnovne tehnike Oracle Clustervare-a koriste funkcije Oracle baze podataka, uključujući usluge, da bi se omogućila sofisticirana funkcionalnost. Na određenim sistemima, Oracle Database i dalje nudi podršku za nekoliko različitih rešenja klastera nezavisnih proizvođača.

Oracle RAC je ciljna publika za Oracle Clustervare, koja je snažno povezana sa njim. Aktivnostima visoke dostupnosti u klasteru može se upravljati pomoću Oracle Clustervare-a. Oracle RAC baza podataka, zajedno sa ostalim neophodnim elementima kao što su adresa virtuelnog internet protokola (VIP), ime za pristup pojedinačnom klijentu (SCAN), SCAN slušalac, Oracle Notification Service i Oracle Net slušaoci, se registruju i njima se upravlja od Oracle Clustervare kada kreirate Oracle RAC bazu podataka koristeći bilo koji od alata za upravljanje. Kada Oracle Clustervare pokrene čvor, ovi resursi se odmah pokreću, a ako ne uspeju, takođe se ponovo pokreću. Na svakom čvoru su aktivni Oracle Clustervare demoni.

CRS resurs je sve čime se bavi Oracle Clustervare. Baza podataka, instanca, usluga, slušalac, VIP adresa ili procedura aplikacije mogu se smatrati CRS resursima. Na osnovu podataka o konfiguraciji za resurs koji se čuva u Oracle Cluster Registri, Oracle Clustervare održava CRS resurse (OCR). Bilo kojim Oracle definisanim CRS resursom može se upravljati pomoću SRVCTL komandi. Infrastruktura koju obezbeđuje Oracle Clustervare omogućava da izgradite CRS resurse za upravljanje bilo kojim procesom koji radi na serverima klastera koji nije unapred definisan od strane Oraclea. Konfiguraciju ovih komponenti predstavljaju podaci koje Oracle Clustervare održava u OCR-u, a kojima možete upravljati prema uputstvima u Oracle Clustervare vodiču za administraciju i primenu.

4.2. O Oracle RAC softverskim komponentama

Oracle RAC baze podataka generalno imaju dve ili više instanci baze podataka od kojih svaka sadrži memorijske strukture i pozadinske procese. Oracle RAC baza podataka ima iste procese i memorijske strukture kao neklasterska Oracle baza podataka i dodatne procese i memorijske strukture koje su specifične za Oracle RAC. Prikaz baze podataka bilo koje instance je skoro identičan pogledu bilo koje druge instance u istoj Oracle RAC bazi podataka; pogled je jedinstvena sistemska slika okoline.

Svaka instanca ima keš bafera u svojoj sistemskoj globalnoj oblasti (SGA). Koristeći Cache Fusion, Oracle RAC okruženja logički kombinuju keš bafera svake instance kako bi omogućili instancama da obrađuju podatke kao da se podaci nalaze u logički kombinovanom, pojedinačnom kešu.

Oracle RAC instance koriste dva procesa, Global Cache Service (GCS) i Global Enkueue Service, kako bi se garantovalo da svaka instanca Oracle RAC baze podataka dobije blok koji joj je potreban za završetak upita ili transakcije (GES). Koristeći Globalni direktorijum resursa, GCS i GES prate status svake datoteke podataka i keširanih blokova (GRD). Sve trenutne instance dele sadržaj GRD-a, čime se proširuje SGA za Oracle RAC instancu.

Nakon keširanja podataka od strane jedne instance, bilo koja druga instanca unutar iste baze podataka klastera može dobiti sliku bloka od druge instance unutar iste baze podataka brže od čitanja bloka sa diska. Kao rezultat toga, umesto da ponovo čita blokove sa diska, Cache Fusion prenosi trenutne blokove između instanci. Cache Fusion prenosi sliku bloka direktno između pogođenih instanci kada je potreban konzistentan blok ili modifikovani blok na drugoj instanci. Oracle RAC komunicira između instanci i prenosi blokove preko privatne interkonekcije. Pristupom Keš Fusion resursima i procedurama oporavka u red čekanja upravljaju GES Monitor i Instance Enkueue Process.

4.3. O Oracle RAC pozadinskim procesima

GCS i GES procesi i GRD sarađuju kako bi omogućili Cache Fusion. Oracle RAC procesi i njihovi identifikatori su sledeći:

- ACMS: Atomic Controlfile to Memori Service (ACMS):U Oracle RAC okruženju, ACMS proces po instanci je agent koji doprinosi obezbeđivanju da je ažuriranje distribuirane SGA memorije ili globalno posvećeno uspehu ili globalno prekinuto ako dođe do kvara.

- GTKSO-j: Globalni proces transakcije: GTKSO-j proces pruža transparentnu podršku za KSA globalne transakcije u Oracle RAC okruženju. Baza podataka automatski podešava broj ovih procesa na osnovu radnog opterećenja KSA globalnih transakcija.
- LMON: Global Enkueue Service Monitor: LMON proces nadgleda globalne redove i resurse u čitavom klasteru i izvodi globalne operacije oporavka čekanja.
- LMD: Global Enkueue Service Daemon: LMD proces upravlja dolaznim zahtevima za udaljenim resursima unutar svake instance.
- LMS: Globalni proces usluge keša: LMS proces održava evidenciju statusa datoteke podataka i svakog keširanog bloka tako što beleži informacije u Globalni direktorijum resursa (GRD). LMS proces takođe kontroliše tok poruka ka udaljenim instancama i upravlja globalnim pristupom blokovima podataka i prenosi slike blokova između keša bafera različitih instanci. Ova obrada je deo funkcije Cache Fusion.
- LCK0: Proces stavljanja instance u red: LCK0 proces upravlja zahtevima za resurse koji nisu Cache Fusion kao što su zahtevi za biblioteku i keš redova.
- RMSn: Oracle RAC procesi upravljanja (RMSn): RMSn procesi obavljaju zadatke upravljivosti za Oracle RAC. Zadaci koje ostvaruje RMSn proces uključuju kreiranje resursa koji se odnose na Oracle RAC kada se nove instance dodaju u klastere.
- RSMN: Remote Slave Monitor upravlja kreiranjem pozadinskog slave procesa i komunikacijom na udaljenim instancama. Ovi pozadinski slave procesi obavljaju zadatke u ime koordinacionog procesa koji se izvodi u drugoj instanci.

4.4. Oracleova arhitektura paralelnog izvršavanja

Arhitektura paralelnog izvršavanja (PKS) iz Oracle-a sastoji se od procesa koordinatora upita (KC) i grupe procesa (PK slave) koji obezbeđuju paralelizam unutar operacije. Oracle podržava paralelno izvršavanje svih relacionih operacija (npr. različita skeniranja, spajanja, poređanja, agregacije, operacije postavljanja), DML (umetanje, ažuriranje, brisanje, spajanje i umetanje više tabela), DDL (npr. kreiranje tabele, indeksa, materijalizovano prikazi), reorganizacija podataka (npr. operacije održavanja particija kao što su premeštanje, razdvajanje, spajanje), učitavanje/isčitavanje preko spoljnih tabela i SQL ekstenzije za Analitiku.

4.4.1. Kako funkcioniraju paralelno izvršavanje

Paralelno izvršavanje deli zadatak izvršavanja SKL naredbe na više malih jedinica, od kojih se svaka izvršava posebnim procesom. Proces senke korisnika koji želi da izvrši upit paralelno preuzima ulogu koordinatora paralelnog izvršavanja ili koordinatora upita (KC). Koordinator upita radi sledeće:

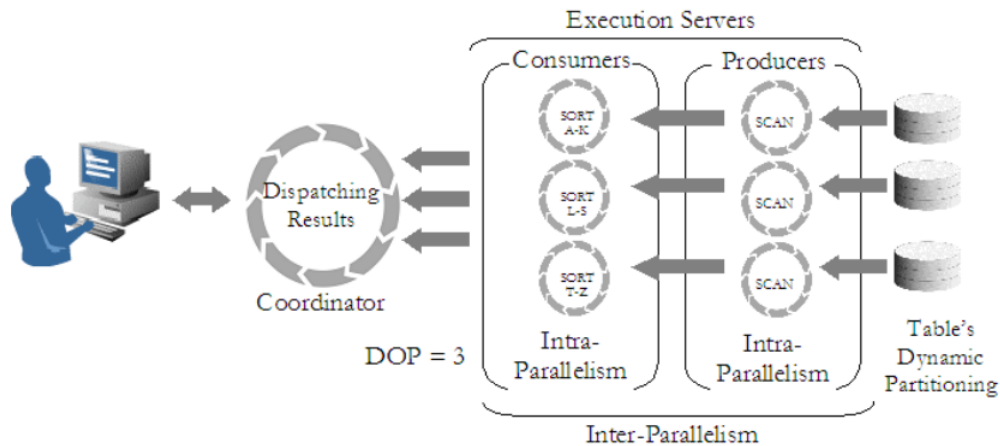
- Parsira upit i određuje stepen paralelizma
- Dodeljuje jedan ili dva skupa servera za paralelno izvršavanje
- Kontrolira upit i šalje uputstva PK slaveovima
- Određuje koje tabele ili indekse treba skenirati
- Proizvodi konačni izlaz korisniku

Paralelni izrazi se izvršavaju na podređenim procesima koji se mogu pokrenuti unutar čvora (paralelizam unutar čvora) ili preko više čvorova (paralelizam među čvorovima) širom mreže. Komunikacija unutar čvora koristi zajedničku memoriju, a komunikacija među čvorovima koristi IPC protokol preko interkonekcija velike brzine. Nakon što je izjava potpuno obrađena, serveri za paralelno izvršavanje se vraćaju na bazen.

4.4.2. Paralelne operacije i stepen paralelizma (DOP)

Nakon što optimizator odredi plan izvršenja naredbe, koordinator paralelnog izvršavanja određuje metod paralelizacije za svaku operaciju u planu izvršenja. Koordinator mora da odluči da li se neka operacija može izvoditi paralelno i, ako može, koliko servera za paralelno izvršavanje da angažuje. Broj servera za paralelno izvršavanje koji se koriste za operaciju je stepen paralelizma (DOP) i određuje se u vreme izvršenja upita, na osnovu trenutnog radnog opterećenja sistema i drugih kriterijuma, kao što je poslovni prioritet upita korisnika.

```
SELECT cust_last_name,cust_first_name  
FROM kupaca ORDER BY cust_last_name;
```



Slika 3. Način rada paralelnih operacija i DOP-a

Plan izvršenja implementira potpuno skeniranje tabele CUSTOMERS nakon čega sledi neka vrsta preuzetih redova na osnovu vrednosti kolone CUST_LAST_NAME. Zarad ovog primera, pretpostavimo da ova kolona nije indeksirana. Takođe pretpostavite da je stepen paralelizma za upit postavljen na tri, što znači da tri paralelna servera za izvršavanje mogu biti aktivna za bilo koju datu operaciju. Svaka od dve operacije (skeniranje i sortiranje) koje se obavljaju istovremeno dobija sopstveni skup servera za paralelno izvršavanje. Dakle, obe operacije imaju paralelizam. Paralelizacija pojedinačne operacije gde se ista operacija izvodi na manjim skupovima redova od strane paralelnih servera za izvršavanje naziva se paralelizam unutar operacije. Kada se dve operacije pokreću istovremeno na različitim skupovima servera za paralelno izvršavanje sa podacima koji teku iz jedne operacije u drugu, postizemo ono što se naziva paralelizam među operacijama. Zbog proizvođačke/potrošačke prirode operacija Oracle servera, samo dve operacije u datom stablu treba da se izvrše istovremeno da bi se vreme izvršenja svelo na minimum.

4.5. Paralelizam, Oracle particionisanje i Oracle RAC

Podrazumevano, sve Oracle RAC instance koje su dostupne se procenjuju za paralelno izvršavanje. Paralelno izvršavanje ne raspoređuje nasumično slave među dostupnim instancama, već počinje sa najmanje učitanom instancom. Cilj je da se smanji saobraćaj između čvorova, istovremeno pokušavajući da minimizira bilo kakvu neravnotežu u radu među instancama.

4.5.1. Koristite particione spojeve

Kada se potpuno particiono spajanje izvršava paralelno, granula paralelizma je particija. Kao rezultat, stepen paralelizma je ograničen na broj particija. Na primer, potrebno vam je najmanje 16 particija da biste postavili stepen paralelizma upita na 16. Različite metode particionisanja se tada mogu koristiti za izjednačavanje obe tabele u ključnoj koloni. Pridruživanja na osnovu particija smanjuju vreme odgovora na upit minimizirajući količinu podataka koji se razmenjuju između paralelnih servera za izvršavanje kada se spajanja izvršavaju paralelno. Ovo značajno smanjuje vreme odziva i poboljšava korišćenje i CPU i memorijskih resursa. U okruženjima Oracle Real Application Clusters, spajanja po particijama takođe izbegavaju ili bar ograničavaju saobraćaj podataka preko interkonekcije, što je ključ za postizanje dobre skalabilnosti za masovne operacije spajanja. Većina operacija spajanja u Oracle Real Application Clusters može imati velike kašnjenja međupovezivanja bez paralelnog izvršavanja particionih spajanja. Trebalo bi da koristite ovu funkciju za velike DSS konfiguracije koje koriste Oracle Real Application Clusters.

4.5.2. Koristite grupe paralelnih instanci (u Oracle bazi podataka 10g)

Oracleov dizajn mreže i optimizacija paralelnih upita će dati dobre performanse u velikoj većini scenarija. Optimizator u sistemu Oracle Real Application Clusters razmatra cenu slanja poruke preko interkonekcije u odnosu na slanje poruke lokalno. Takođe uzima u obzir broj aktivnih instanci. Uprkos činjenici da će optimizator pokušati da izvrši upit na jednoj instanci... Kao rezultat toga, ako se očekuje da će upit vratiti veliki broj redova iz svakog čvora, možda bi bilo korisno ograničiti paralelizam među čvorovima kao što je gore navedeno, čime se ograničava količina podataka koji se prenose preko interkonekcije. Takođe, ako svaki čvor vraća samo ograničen broj redova, možda bi bilo bolje da se minimizira broj čvorova koji učestvuju zbog troškova vremena udaljenog pokretanja procesa.

Način da se minimizira saobraćaj između čvorova je da se ograniči paralelno izvršavanje unutar jedne instance ili grupe instanci. Možete uspostaviti članstvo u grupi postavljanjem parametara `INSTANCE_GROUPS` i `PARALLEL_INSTANCE_GROUP`.

Na primer, uzmite u obzir sledeće dodele:

Na instanci A: `INSTANCE_GROUPS=AMER`

Na instanci B: `INSTANCE_GROUPS=AMER, EMEA, APAC, JPN`

Zatim korisnik može da aktivira čvorove u grupi AMER da bi pokrenuo procese servera upita koristeći sledeću komandu: ALTER SESSION SET PARALLEL_INSTANCE_GROUP = AMER;

Kao odgovor, paralelna izvršenja se mogu podeliti na instance, tj. mogu se izvoditi i na instanci A i na instanci B. S druge strane, postavljanjem PARALLEL_INSTANCE_GROUP = APAC, samo instanca B se može koristiti za paralelna izvršenja. Imajte na umu da parametar init.ora INSTANCE_GROUPS ne može da se menja dinamički.

Napomena: U Oracle Database 11g paralelno izvršavanje je sada svesno definicije usluge i automatski preuzima odgovarajuću postavku PARALLEL_INSTANCE_GROUP, čime izričito podešavanje PARALLEL_INSTANCE_GROUP čini nepotrebnim. Korišćenje usluga će biti objašnjeno u odeljku za upravljanje radnim opterećenjem.

Parallel_adaptive_multi_user Parametar

PARALLEL_ADAPTIVE_MULTI_USER, kada je postavljeno na TRUE, omogućava adaptivni algoritam dizajniran da poboljša performanse u višekorisničkim okruženjima koja koriste paralelno izvršavanje. Algoritam automatski smanjuje traženi stepen paralelizma na osnovu opterećenja sistema u vreme pokretanja upita. Efektivni stepen paralelizma je zasnovan na podrazumevanom stepenu paralelizma, ili stepenu tabele ili nagoveštaja, podeljenom sa faktorom smanjenja.

Primer: Na mašini sa 17 CPU podrazumevani stepen paralelizma može biti postavljen na 32. Ako jedan korisnik postavi paralelni upit, taj korisnik dobija stepen od 32, efektivno koristeći sve CPU i memoriju u sistemu. Kada drugi korisnik uđe u sistem, taj korisnik dobija stepen od 16. Kako se broj korisnika u sistemu povećava, ovaj algoritam će nastaviti da smanjuje stepen sve dok korisnici ne budu pokrenuli koristeći stepen 1, kada ima 32 korisnika na sistemu sistem. Podrazumevana vrednost parallel_adaptive_multi_user je tačna.

4.5.3. Razmatranja o međusobnom povezivanju

Međuprocesna komunikacija čini većinu interkonektivnog saobraćaja u sistemima skladišta podataka (IPC). Paralelni serveri koriste interkonekciju za međusobnu komunikaciju. Obim interkonektivnog saobraćaja zavisi od operacije i broja uključenih čvorova. Zbog mogućnosti povezivanja preko paralelnih servera, procedure pridruživanja obično zahtevaju više interkonektivnog saobraćaja od običnih agregacija. U zavisnosti od tehnike distribucije, obim interkonektivnog saobraćaja može dramatično da varira. Mehanizam distribucije koji se

koristi može se otkriti u polju DISTRIBUCIJA plana upita. Najveći saobraćaj interkonekcije se generiše kada se emituje jedna strana spajanja ili se obe strane distribuiraju heš. Particionisanje u delovima Spajanja sa samo jednostranom redistribucijom rezultiraju manje saobraćaja međupovezivanja, ali puna particiona spajanja bez bočne preraspodele rezultiraju najmanje saobraćaja međupovezivanja. Kao rezultat toga, ako se izvede paralelni upit između čvorova, mreža za povezivanje mora biti adekvatno skalirana. Osim ako aplikacija nije veoma dobro strukturirana, sa uglavnom unapred definisanim skupom upita koji mogu imati koristi od spajanja po particijama, imajte na umu da će se određena redistribucija podataka desiti preko interkonekcije za istovremene pretrage među čvorovima.

Na obim interkonektivnog saobraćaja utiče i broj čvorova uključenih u proceduru spajanja. Što je više čvorova uključeno u proces pridruživanja, to se više podataka mora poslati udaljenim čvorovima. Na primer, u Oracle RAC klasteru sa 4 čvora sa po 4 CPU-a da bi se maksimizirale performanse učitavanja sa eksternim tabelama, potrebno je podesiti DOP na 32 i na spoljnim i na internim tabelama. Ovo će dovesti do toga da 8 paralelnih servera obavljaju operacije čitanja iz spoljne tabele na svakom čvoru, kao i da 8 paralelnih servera izvršavaju izjave za kreiranje tabele na svakom čvoru.

Oracle RAC mrežni saobraćaj se distribuira između navedenih IP adresa.
`CLUSTER_INTERCONNECTS = ip1:ip2:...:ipn`

Oracle koristi sve interkonekcije koje su navedene. Ovo obezbeđuje balansiranje opterećenja sve dok sve navedene interkonekcije ostaju u funkciji.

Imajte na umu da ako jedna od interkonekcija u parametru `CLUSTER_INTERCONNECTS` postane nedostupna, Oracle vraća grešku i instanca možda neće uspeti. Napomena: Ako koristite parametar `CLUSTER_INTERONNECTS` za balansiranje opterećenja i performanse, to će biti po ceni visoke dostupnosti. Razmislite o povećanju parametra `PARALLEL_EKSECUTION_MESSAGE_SIZE`. Ovaj parametar određuje veličinu bafera koji se koristi za poruke o paralelnom izvršavanju. Podrazumevana vrednost je obično 2K. Povećanje ove vrednosti (na 4K ili 8K na primer) može poboljšati performanse.

Slede 2 primera AVR izveštaja koji naglašava 2 primarna tipa saobraćaja međupovezivanja:

Globalni keš saobraćaj (tipično za OLTP operacije)

Global Cache Load Profile	Per Sec	Per Trans
-----	-----	-----
Global Cache blocks received:	2.70	2.23
Global Cache blocks served:	2.84	2.36
GCS/GES messages received:	164.07	136.03
GCS/GES messages sent:	136.96	113.56
DBWR Fusion writes:	0.22	0.18

Slika 4. Primer za globalni keš saobraćaj

Statistic	Total	per Sec	per Trans
-----	-----	-----	-----
PX local messages recv'd	104	0.1	0.1
PX local messages sent	104	0.1	0.1
PX remote messages recv'd	200271	200.2	151.1
PX remote messages sent	213267	213.2	156.1

Slika 5. Primer za IPK saobraćaj (za paralelne operacije)

4.6. Automatsko upravljanje opterećenjem

Automatsko upravljanje opterećenjem automatizuje i reguliše distribuciju posla između čvorova klastera kako bi se obezbedile maksimalne performanse za korisnike i aplikacije.

4.6.1. Usluge

Oracle Database 10g uvodi usluge, snažnu sposobnost upravljanja radnim opterećenjem. Oni služe kao osnova za upravljanje radnim opterećenjem u Oracle RAC-u. Radno opterećenje Oracle baze podataka je podeljeno u grupe koje se međusobno isključuju prema uslugama. Svaka usluga predstavlja radno opterećenje/aplikaciju sa zajedničkim

svojstvima, pragovima nivoa usluge, prioritetom, metrikom učinka u realnom vremenu i upozorenjima i radnjama kada ciljevi učinka nisu ispunjeni. Kada pravite uslugu, navodite koje instance će je normalno podržavati. Ovo su poželjne okolnosti za tu uslugu. Druge instance se također mogu definisati da pružaju uslugu ako primarna instanca ne uspe. Oni se nazivaju dostupnim instancama usluge. Kada definišete željenu instancu za uslugu, ta instanca se koristi tokom normalnog rada. Oracle Clustervare čini sve da garantuje da usluga uvek radi na svim željenim instancama koje su za njega podešene. Ako instanca ne uspe, usluga se nasumično premešta u jednu od dostupnih instanci. Takođe možete ručno da premestite uslugu na dostupnu instancu. Kada uspostavite uslugu i ne navedete željene ili dostupne instance, podrazumevano je svaka instanca u Oracle RAC bazi podataka preferirana instanca za tu uslugu.

Primer: `srvctl add service -d rac_db -s dv_load -r inst1,inst2 -a inst3,inst4`
`srvctl add service -d rac_db -s dv_kueri -r inst3,inst4 -a inst1,inst2`

U ovom primeru su kreirane dve usluge, DV_LOAD i DV_KUERI. Usluga DV_LOAD koristi instance inst1 i inst2 kao svoje preferirane instance sa inst3 i inst4 kao dostupne instance. Usluga DV_KUERI koristi obrnutu od ovih dodela. Aplikacije srednjeg nivoa i aplikacije klijent-server koriste uslugu navodeći uslugu kao deo veze u podacima TNS povezivanja. Ovo može biti u datoteci TNSnames za drajvere za gustu mrežu, u URL specifikaciji za tanke drajvere ili se može održavati u Oracle Internet direktorijumu. Primer tnsnames.ora: `LOAD_USERS = (DESCRIPTION = (LISTA ADRESA = Usluga (ADRESA = (PROTOKOL = TCP)(HOST = docrac1-vip)(PORT = 1521)) (ADRESA = (PROTOKOL = TCP)(HOST = docrac2-vip)(PORT = 1521)) (LOAD_BALANCE = da)) (CONNECT_DATA = (SERVICE_NAME = DV_LOAD)))`

Rad na strani servera, kao što su Planer poslova, Parallel Kueri i Oracle Streams Advanced Kueuing postavljaju ime usluge kao deo radnog opterećenja definicija. Za Planer poslova, usluga koju koristi klasa poslova je definisana kada se kreira klasa posla. Tokom izvršenja, poslovi se dodeljuju klasama poslova i klase poslova se pokreću unutar usluga. Korišćenje usluga sa klasama poslova osigurava da je rad planera poslova identifikovan za upravljanje radnim opterećenjem i podešavanje performansi. Za visoku dostupnost, DBMS_JOBS je prethodno koristio ime instance da definiše gde bi usluge trebalo da se izvršavaju. Ovaj pristup je doveo do toga da poslovi nisu pokrenuti kada instanca nije bila dostupna. Sa novim planerom poslova, postavljanje usluge u klasu poslova osigurava da se posao izvršava kada je usluga pokrenuta bilo gde u klasteru ili mreži.

ZAKLJUČAK

Aplikacije za analizu podataka i rudarenje podataka u velikoj meri se oslanjaju na grupisanje. Kako se skupovi podataka povećavaju i njihove karakteristike i međusobne veze menjaju, posebno je poželjna mogućnost da se otkriju visoko povezane oblasti stavki kada njihov broj postane izuzetno ogroman. Bez obzira na to, svako klasterisanje „je klasifikacija stavki u grupe na osnovu skupa kriterijuma – nije ni tačno ni netačno“, prema autorima. Iz perspektive rudarenja podataka, u ovoj studiji smo razmatrali proceduru grupisanja. Razmotrene su karakteristike „dobre“ metodologije klasterisanja, kao i strategije za lociranje relevantnih particija. U isto vreme, došli smo do zaključka da se istraživanje fokusiralo na numeričke skupove podataka, ostavljajući složenost rada sa ogromnim bazama podataka kategorija na nekoliko alternativnih metodologija. Utvrdili smo da su potrebna nova istraživačka rešenja za kategoričko grupisanje podataka i dali smo svoje predloge za buduća proučavanja.

Kako su moderna skladišta podataka i njihovi tehnički zahtevi postali sve češći i složeniji, oni su počeli da zahtevaju veću fleksibilnost, veću dostupnost i još bolje performanse svoje infrastrukture baze podataka. Ovi sve veći zahtevi mogu se lako ispuniti implementacijom Oracle Real Application Clusters (RAC). Ova arhitektura, u kombinaciji sa jakim karakteristikama Oracle RDBMS-a, može da obezbedi stabilnost, dostupnost, skalabilnost i opštu funkcionalnost kako bi sistem skladišta podataka bio uspešan. Pored toga, uspeh sistema skladišta podataka implementiranog na Oracle RAC zavisi od ravnoteže: ravnoteže u dizajnu između optimizacija za poslovne potrebe naspram operativnih potreba, ravnoteže u implementaciji za performanse nekoliko slučajnih korisničkih pristupa naspram skalabilnih performansi za zadovoljiti bilo koji broj zahteva za istovremenu obradu. Na kraju, sistem mora biti dizajniran i implementiran tako da na najbolji način zadovolji potrebe poslovanja dok istovremeno svede operativne zahteve na minimum.

LITERATURA

1. Ferligoj, A.: Cluster analysis, 2002. materijali sa School of Biometrics, Cavtat, June 25, 2002.
2. Hastie, T., R. Tibshirani i J.H. Friedman: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, 2001.
3. Periklis, A.: Data Clustering Techniques. Rapport technique, University of Toronto, Department of Computer Science, 2002.
4. Uncovering excellence in cluster management, PWC, February 2011
5. Porter, M.E., "On Competition", Harvard Business Press, 1998
6. Porter, M.E., "The Competitive Advantage of Nations", Free Press, New York, 1990. (1990, sec.ed. 1998)
7. Michael R. Anderberg. Cluster analysis for applications. Academic Press, 1973.
8. Brian S. Everitt. Cluster Analysis. Edward Arnold, 1993.
9. Leonard Kaufman and Peter J. Rousseeuw. Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley & Sons, 1990.
10. <https://developer.ibm.com/articles/os-weka2/>
11. E. W. Gilbert. Pioneer Maps of Health and Disease in England. Geographical Journal, 124: 172– 183, 1958.
12. Anil K. Jain and Richard C. Dubes. Algorithms for Clustering Data. Prentice-Hall, 1988.
13. Data Clustering Techniques Qualifying Oral Examination Paper Periklis Andritsos University of Toronto Department of Computer Science, March 11, 2002
14. Jiawei Han and Michelle Kamber. Data Mining: Concepts and Techniques. Morgan Kaufmann, 2001.
15. <http://www.census.gov/>
16. https://docs.oracle.com/cd/E11882_01/rac.112/e41960/admcon.htm?fbclid=IwAR0RFO5IGremCRu1iU4EfudQiCI8F0hsj-ZyfC76SvbBCfDdLfQIVOW2eUY#RACAD7276
17. https://docs.oracle.com/cd/E11882_01/rac.112/e41960/admcon.htm?fbclid=IwAR0RFO5IGremCRu1iU4EfudQiCI8F0hsj-ZyfC76SvbBCfDdLfQIVOW2eUY#RACAD7276
18. Data Warehousing on Oracle RAC Best Practices An Oracle White Paper October 2008
19. Data Warehousing on Oracle RAC Best Practices An Oracle White Paper October 2008