FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVOD ZA ELEKTRONIČKE SUSTAVE I OBRADBU INFORMACIJA

SUSTAVI ZA PRAĆENJE I VOĐENJE PROCESA

DISTRIBUIRANE BAZE PODATAKA

Seminar

Šarčević Konstantin
0036389474

Svibanj, 2006.

Uvod

Ako računalo u računalnoj mreži definiramo kao mozak organizacije, onda je mreža živčani sustav organizacije, a baza podataka kolektivno pamćenje organizacije.

Definicija koju daje James Martin:

„Baza podataka je skup istovrsnih podataka s višestrukom namjenom. Korisnik nije zainteresiran za sve vrste podataka u bazi, već samo za one koji su mu potrebni u njegovu poslu. Korisnik može imati uvid u samo jednu, njemu potrebnu datoteku, koja ima uvijek istu i to vrlo jednostavnu strukturu, iako je u biti izvedena iz mnogo kompleksnije strukture podataka. Različiti korisnici uzimaju u obzir različite datoteke izvedene iz iste baze podataka. Dakle, iako je baza podataka zajednička većem broju korisnika, različiti korisnici je različito shvaćaju."

Prema Ulmanu, zajednička osobina svih baza podataka je:

- Apstraktni model podataka
- Visoka razina pristupa ili upitnih jezika
- Upravljanje transakcijama u višekorisničkom okruženju
- Kontrola pristupa i vlasništvo nad podacima
- Validacija podataka i provjera konzistentnosti
- Konzistentni oporavak sustava nakon ispadanja sustava i/ili strojne opreme

Baze podataka se dijele na:

Operacijske (transakcijske, produkcijske) baze podataka - pamte detaljne i aktuelne podatke potrebne za podršku poslovnih procesa i operacija u e-biznis preduzeću. Primjeri su baza kupaca, baza ljudskih resursa – personala, baza zaliha i mnoge druge baze generisane poslovnim operacijama.

Distribuirane baze podataka - Mnoge organizacije repliciraju i distribuiraju kopije ili djelove baze podataka na različite mrežne servere. Distribuirane baze dakle se mogu nalaziti na različitim serverima Web, intranet, extranet ili neke druge kompanijske mreže. Distribuirane baze mogu biti kopije operacijskih ili analitičkih, hipermedijalnih ili nekih drugih tipova baza podataka. Replikacija i distribucija baza podataka umanjuje performanse i sigurnost baze podataka. Garancija da će svi podaci u organizacijskim distribuiranim bazama podataka biti konzistentno i istodobno ažurirani je jedan od glavnih zadataka distribuiranih DBMS-a.

Eksterne baze podataka - Pristup informacijama iz eksternih baza podataka je moguć putem velikog broja komercijalnih online servisa, kao i sa plaćanjem ili bez, iz mnogih izvora na Internetu. Web-stranice osiguravaju putem hiperlinkovanih strana pretraživanje multimedijalnih dokumenata u hipermedijalnim bazama podataka. Eksterni podaci su najčešće u vidu statističkih baza podataka nastalih kao rezultat ekonomskih i

demografskih istraživanja i aktivnosti. Također su popularne bibliografske ili full text baze iz kojih se mogu vidjeti ili download-ovati apstrakti ili kompletne kopije raznih časopisa, istraživačkih radova i drugih publiciranih materijala.

Data Warehouse i Data Mining - Data Warehouse baza podataka memorizira podatke ekstrahirane iz različitih operacijskih, eksternih ili nekih drugih baza podataka u okviru organizacije. Ona je središnji izvor prečišćenih i transformiranih aktuelnih i povijesnih podataka koje koriste menadžeri i drugi poslovni stručnjaci za data mining, OLAP (online analytical processing) i druge vrste poslovnih analiza, istraživanja tržišta, i podršku odlučivanju. Data warehouse može biti podijeljen na **data mart-**ove koji sadrže podskupove podataka warehose-a koji su fokusirani na neki specifični aspekt kompanije, kao što je odjel ili poslovni proces.

Hipermedijalne baze podataka na Web-u. Rapidni napredak Web tehnologija je značajno povećao primjenu baza hipermedijalnih dokumenata koje se nazivaju hipermedijalne baze podataka. Hipermedijalna baza podataka je zapravo webstranica koja se sastoji od hiperlinkovanih multimedijalnih strana. Dakle, sa točke gledišta teorije baza podataka, hipermedijalna baza podataka nije skup međusobno povezanih slogova, već skup međusobno hiperlinkovanih multimedijalnih strana.

Distribuirane baze podataka

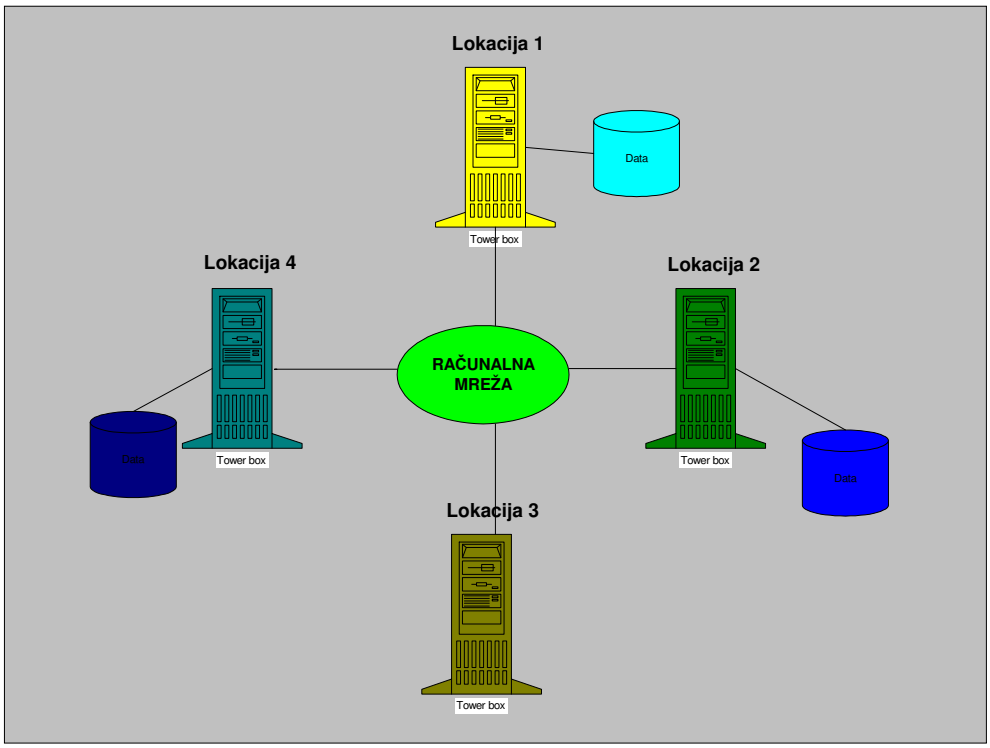
Distribuirana baza podataka je baza podataka koja se ne nalazi u cjelini na jednom računalu, već je razdijeljena na više lokacija koje su povezane komunikacijskom mrežom. Svaka lokacija, koja se zove i čvor komunikacijske mreže, ima svoj vlastiti, autonomni sustav za upravljanje bazama podataka, sa vlastitom kontrolom, upravljačem transakcija i opravka od pada, i ostalim važnim funkcijama, a ima i svoj procesor i ulazno/izlazne uređaje.

Aplikacije istovremeno pristupaju i mijenjaju podatke na više različitih baza podataka u mreži, gdje mreža može biti LAN ili WAN.

Osnovne značajke DBP:

- Skup logički povezanih djeljivih podataka
- Podaci su razdvojeni na više fragmenata
- Fragmenti se mogu replicirati
- Fragmenti/Replikacije pripadaju lokacijama
- Lokacije su povezane komunikacijskom mrežom
- Podaci na svakoj lokaciji su pod nadzorom DBMS-a
- DBMS na svakoj lokaciji može upravljati lokalnim aplikacijama autonomno
- Svaki DBMS učestvuje u najmanje jednoj globalnoj aplikaciji.

Prikaz sustava DBP:



Prednosti koje donosi distribuirani koncept baze podataka:

- **Lokalna autonomija podataka, upravljanja i kontrole**. Okruženje u kojem se DBP primjenjuju je i samo distribuirano (npr. sveučilište, fakulteti, zavodi; središnja knjižnica, matične knjižnice, ogranci; ministarstvo, regionalni centri itd.).Distribuiranje baza podataka kao i sustava za upravljanje njima, omogućava pojedinim grupama da lokalno kontroliraju vlastite podatke, uz mogućnost pristupa podacima na drugim lokacijama kada je to potrebno.

- **Veći kapacitet i postupni rast**. Čest razlog za instaliranje distribuiranog sustava je nemogućnost jednog računala da primi i obrađuje sve potrebne podatke. U slučaju da potrebe nadmaše postojeći kapacitet, dodavanje čvora distribuiranom sustavu je znatno jednostavnije nego zamjena sustava većim.

- **Pouzdanost i raspoloživost**. Distribuirani sustavi mogu nastaviti svoje funkcioniranje i kada neki od čvorova izgube funkcionalnost.

- **Efikasnost i fleksibilnost**. Podaci su fizički blizu onome tko ih stvara i koristi, pa je znatno smanjena potreba za udaljenom komunikacijom.

Da bi se postigla visoka efikasnost DSUBP, sve njegove komponente treba realizirati na takav način da se smanji mrežna komunikacija. Najznačajniji problemi koje pri tome treba riješiti su:

- fragmentacija podataka
- distribuirana obradba upita
- distribuirano ažuriranje
- upravljanje katalogom
- distribuirano izvršenje skupa transakcija, što uključuje konkurentnost, integritet, oporavak i protokole kompletiranja transakcija

Fragmentacija podataka

Podaci u distribuiranom sustavu mogu biti particionirani ili ponovljeni u fizičkoj memoriji.

U slučaju ponavljanja podataka, jedan logički objekt može imati veći broj kopija na većem broju lokacija. Ponovljenost podataka povećava raspoloživost podataka i efikasnost pristupa podacima, ali značajno komplicira ažuriranje podataka, koji moraju biti konzistentni u svim svojim kopijama.Složenost koju nosi sa sobom strategija ponavljanja podataka mora biti sakrivena od korisnika, tj. mora biti osigurana nevidljivost ponavljanja podataka.

U slučaju particioniranja podataka,logički skup podataka treba na neki način podijeliti, a zatim dijelove - fragmente (može i sa ponovljenim kopijama) razdijeliti po raznim lokacijama. Logički skup podataka u relacijskom sustavu je relacija, a prirodni fragment relacije je neki njezin podskup definiran uvjetom projekcije i restrikcije. Fragmentacija

mora biti izvedena tako da se spajanjem fragmenata dobije polazna relacija.

Distribuirana obradba upita

Distribuirana obradba upita podrazumijeva distribuiranu optimizaciju kao i distribuirano izvršenje upita. Strategije optimizacije upita nad distribuiranom bazom podataka imaju za cilj minimalizaciju cijene obradbe i vremena za koje će korisnik dobiti odgovor. U troškovima obradbe najveći udio imaju troškovi mrežne komunikacije, tj. prijenosa podataka kroz mrežu, dok su troškovi komunikacije sa ulazno/izlaznim uređajima i korištenja procesora manji za nekoliko redova veličine. Zbog toga je vrlo značajno, u ovisnosti o propusnosti mreže (količina podataka koju može primiti u sekundi) i vremena kašnjenja, pravilno odabrati relacije i njihove fragmente koji će biti prenošeni s jedne lokacije na drugu s ciljem obradbe upita (globalna optimizacija). Razlog za prenošenje podataka može biti taj što su podaci na lokaciji različitoj od one na koju se postavlja upit, ili što u upitu učestvuje veći broj relacija s različitih lokacija. Izbor strategije za izvršenje operacija na jednoj lokaciji zove se lokalna optimizacija.

Ako se n relacija R_1 , R_2, …,R_n koje učestvuju u upitu nalaze na k različitih lokacija l_1 , l_2 , …, l_n , pri čemu je svaka relacija R_i u cijelosti na lokaciji l_j , onda se osnovna strategija distribuirane obradbe upita sastoji od dva koraka:

1. maksimalna redukcija svake relacije na njenoj lokaciji (lokalna restrikcija i projekcija na atribute spajanja i izlazne atribute)
2. prenošenje dobivenih relacija na jednu lokaciju, ili na više lokacija, redom, na kojima je moguće izvršiti pojedinačna spajanja i projekciju na izlazne atribute

Za drugi korak osnovne strategije vezana je odluka o tome koje se relacije prijenose i na koje lokacije. Ta odluka se donosi na temelju procjene količine podataka koji se prijenose između lokacija u svakom pojedinačnom slučaju; izbor relacija i lokacija vrši se tako da se minimizira protok podataka kroz mrežu.

Distribuirano ažuriranje

Kao što je rečeno u dijelu o fragmentaciji podataka, ponavljanje podataka podrazumijeva da jedan logički objekt (relacija ili fragment) može imati više kopija na većem broju lokacija. Posljedica toga je da se, s obzirom na potrebu za konzistentnošću podataka u svim kopijama, ažuriranje jednog logičkog objekta mora prenijeti i na sve fizičke kopije tog objekta. Međutim, trenutno prenošenje ažuriranja može onemogućiti (ili nedopustivo dugo odložiti) uspješno izvršavanje ažuriranja u slučaju da je bilo koja lokacija u padu; na taj način ponavljanje podataka ne povećava, nego smanjuje raspoloživost podataka.

Jedan široko prihvaćeni pristup prijenosu ažuriranja oslanja se na koncept primarne kopije, i sastoji se od dva dijela:

- jedna kopija svakog ponovljenog objekta proglašava se primarnom kopijom tog objekta, pri čemu primarne kopije različitih objekata mogu biti ba različitim lokacijama
- operacija ažuriranja objekta smatra se logički izvršenom čim se izvrši ažuriranje primarne kopije tog objekta; ažuriranje ostalih kopija je sada u nadležnosti lokacije na kojoj je primarna kopija, ali se mora izvršiti prije kompletiranja transakcije. Ovaj postupak zahtijeva primjenu protokola dvofaznog kompletiranja transakcije (bit će opisan kasnije), koji se ne može uspješno provesti ako je bar jedna relevantna kopija u padu , što je čest slučaj. Ukoliko se dopusti ažuriranje kopija i poslije kompletiranja transakcije, ne može se jamčiti konzistentnost podataka u svim njihovim kopijama.

Upravljanje katalogom

Katalog je baza podataka koja sadrži podatke o baznim relacijama, pogledima, indeksima korisnicima itd., a u slučaju distribuiranog sustava, i o načinu i lokacijama na koje su podaci razdijeljeni i (možda) ponovljeni. Sam katalog u distribuiranom sustavu može biti centraliziran (samo na jednoj lokaciji), potpuno ponovljen (na svim lokacijama po jedna kopija kataloga), particioniran (na svakoj lokaciji je dio kataloga koji se odnosi na objekte te lokacije) ili kombiniran (katalog je particioniran ali na jednoj lokaciji postoji jedna središnja kopija cijelog kataloga).

S obzirom na nedostatke koje ima svaki od navedenih pristupa (zavisnost od središnje lokacije, visoka cijena prijenosa ažuriranja kataloga, skup pristup udaljenoj lokaciji) implementirani sustavi koriste neke druge strategije. Npr. u sustavu R* svaka lokacija sadrži:

- slog kataloga za svaki objekt „roden“ na toj lokaciji (tj. čija je prva kopija kreirana na toj lokaciji); ovaj slog sadrži i informaciju o trenutnoj lokaciji objekta
- slog kataloga za svaki objekt koji je trenutno smješten na toj lokaciji
- tabelu sinonima za svakog korisnika prijavljenog na toj lokaciji, koja preslikava sinonime objekta, definirane iskazom kreiranja sinonima, u sistemske identifikatore objekta, jedinstvene u cijelom distribuiranom sustavu

Sistemski identifikator objekta, koji se nikad ne mijenja (dok se objekt ne ukloni) sastoji se od identifikatora korisnika koji je kreirao prvu kopiju tog dokumenta, lokacije sa koje je kreirana, lokalnog imena objekta i lokacije na kojoj je rođen. Pronalaženje objekta kada je zadano njegovo lokalno ime ili sinonim, počinje preslikavanjem lokalnog imena (automatski), odnosno sinonima (pregledavanjem tabele sinonima), u sistemski identifikator objekta. Zatim se, prema sistemskom identifikatoru, nalazi lokacija rođenja objekta, pristupa joj se i u slogu koji odgovara tom objektu, pronalazi se lokacija u kojoj je trenutno smješten. U slijedećem koraku pristupa se lokaciji na kojoj je objekt smješten, i lokalnim operacijama pronalazi objekt.

Distribuirano upravljanje transakcijama

Pod transakcijom u distribuiranom sustavu podrazumijeva se vremenski niz uređenih radnji koji prevodi jedno konzistentno stanje baze u drugo konzistentno stanje. Transakcija može izvršavati može izvršavati radnje svog programa na raznim lokacijama. Na taj način transakcija može izvršavati više procesa na više lokacija. Startanjem jedne transakcije bira se jedan upravljač transakcija (na jednoj lokaciji) koji služi kao koordinator procesa te transakcije. Svakoj transakciji se dodjeljuje i njezin privatni radni prostor (koji može biti raspoređen na više lokacija), iz kojeg će transakcija čitati i u kojeg će se upisivati vrijednosti objekata.

Pri distribuiranoj obradbi transakcija, transakcija koja čita vrijednost objekta, čita vrijednost samo jednog lokalnog primjerka tog objekta, dok ažuriranje jednog objekta sprovodi nad svim primjercima tog objekta.

Konkurentnost

Neka je dat skup transakcija $\{ T_i \}_{i=1}^{n}$ nad bazom podataka distribuiranom nad k lokacija:

Lokalno izvršenje skupa T na lokaciji l je niz I_l trojki oblika:

$(T_j$, pročitaj, x_l), odnosno

$(T_j$, ažuriraj, x_l),

gdje je $T_j \in$ T, x_l je primjerak objekta x na lokaciji l, a radnja (pročitaj x) odnosno (ažuriraj x) je radnja transakcije T_j. Poredak trojki u ovom nizu odgovara poretku radnj pojedine transakcije.

Distribuirano izvršenje skupa T je niz lokalnih izvršenja I=$\{ I_l \}_{l=1}^{k}$, takav da važi:

a) (čitanje samo jednog primjerka): ako je $(T_i$, pročitaj, $x_j) \in I_j$, tada $(T_i$,pročitaj, $x_l) \notin I_l$, za l≠j, gdje su x_j, x_l primjerci objekta x na lokacijiama j,l, redom
b) (ažuriranje svih primjeraka): ako je $(T_i$, ažuriraj, $x_j) \in I_j$, tada je $(T_i$,ažuriraj, $x_l) \in I_l$, za sve lokacije l, na kojima postoji primjerak x_l objekta x
c) svakoj radnji svake transakcije T_j odgovara bar jedna trojka sa prvom komponentom T_j

Kao kod centraliziranih sustava, i kod distribuiranih sustava mogu se definirati serijska distribuirana izvršenja, ekvivalentna distribuirana i linearizirana (ekvivalentna serijskim) distribuirana izvršenja skupa transakcija.

Serijsko distribuirano izvršenje skupa transakcija T=$\{ T_i \}_{i=1}^{n}$ je distribuirano izvršenje I za koje se na skupu T može definirati uređenje < za koje važi: ako je $T_i < T_j$, onda sve radnje transakcije T_i prethode svim radnjama transakcije T_j u svakom lokalnom izvr-

šenju I_l u kojem se pojavljuju radnje obiju transakcija.

Distribuirano izvršenje može ne biti serijsko ili zbog toga što neko od lokalnih izvršenja nije serijsko, ili zbog toga što redoslijed transakcija u (inače serijskim) lokalnim izvršenjima nije isti.

Dvije radnje skupa transakcija T su konfliktne u distribuiranom izvršenju ako su konfliktne u nekom (bilo kom) izvršenju.

Dva distribuirana izvršenja su ekvivalentna ako su, za svaku lokaciju, njihova lokalna izvršenja ekvivalentna (u centraliziranom smislu).

Distribuirano izvršenje je linearizirano ako je ekvivalentno nekom serijskom izvršenju.

Dvofazno zaključivanje

Jedna metoda za postizanje lineariziranog distribuiranog izvršenja je, kao i kod centraliziranog lineariziranog izvršenja, dvofaznost transakcija, koje se u distribuiranom slučaju proširuje nekim zahtjevima:

- pri čitanju logičkog objekta x, dovoljno je da transakcija zaključa djeljivim lokotom jedan primjerak objekta x (onaj koji stvarno čita)
- pri ažuriranju logičkog objekta x, transakcija mora postaviti lokote na sve primjerke objekta x, na svim lokacijama u distribuiranoj bazi
- ako transakcija ne može dobiti traženi lokot, ona stoji u redu za čekanje za onaj primjerak objekta nad kojim treba izvršiti radnju

Glavni nedostatak prethodnog postupka dvofaznog zaključivanja je što transakcija, pri ažuriranju jednog logičkog objekta, mora dobiti lokot na svim primjercim tog objekta, na raznim lokacijama, a o dodjeljivanju i oslobađanju lokota odlučuju lokalni moduli(upravljači zaključivanja). Zbog toga taj postupak zahtijeva mnogo komunikacije (slanja i primanja poruka), i može se pojednostavniti na nekoliko načina (od kojih svaki ima svoje prednosti i nedostatke). Jedna tehnika dvofaznog zaključivanja u distribuiranom sustavu je centralizacija upravljača zaključivanja koja podrazumjeva da samo jedna lokacija upravlja postavljanjem i oslobađanjem lokota u cijelom sustavu. U tom slučaju je ažuriranje pojednostavljeno, jer se skup svih primjeraka jednog logičkog objekta može tretirati (u svrhe zaključivanja) kao jedinstveni objekt. Nedostatak ove tehnike je što se lokacija koja upravlja zaključivanjem brzo opterećuje, a u slučaju njenog pada, pada i cijeli distribuirani sustav.

Druga tehnika je tehnika primarnog primjerka objekta. U ovoj tehnici svaki logički objekt ima jedan svoj primjerak i, moguće, veći broj ostalih primjeraka. Razni logički mogu imati primarne primjerke na raznim lokacijama. Upravljač zaključivanja lokacije na kojoj je primarni primjerak logičkog objekta x obrađuje zahtijeve za zaključivanjem objekta x (i na svim drugim lokacijama). I u ovom slučaju svi primjerci jednog objekta ponašaju se, u svrhe zaključivanja, kao jedinstveni objekt, ali ova tehnika ne pokazuje nedostatke centraliziranog upravljača zaključivanja.

Vremenske oznake

Kod centraliziranih sustava spriječavanje uzajamnog blokiranja je bila skupa operacija pa se naječešće pribjegavalo njegovom detektiranju i poništavanju nekih transakcija koje su u njemu imale učešće. Kod distribuiranih sustava operacija poništavanja transakcija je

skuplja nego kod centraliziranih sustava, pa se primjenjuje i druga mogućnost riješavanja problema uzajamnog blokiranja – njegovo spriječavanje. Ono se postiže na račun smanjene konkurentnosti izvršenja skupa transakcija.

Jedna metoda konkurentnog izvršavanja skupa transakcija koja isključuje mogućnost uzajamnog blokiranja je metoda vremenskih oznaka. U ovoj metodi transakcijama se dodijeljuju oznake vremena njihovog početka, i dopuštaju se konfliktne radnje samou redoslijedu vremenskih oznaka pripadnih transakcija (pokušaj da se ide drugim redoslijedom rezultira poništenjem transakcije). Dok metoda zaključivanja, uz detekciju i razriješavanje nastalog uzajamnog blokiranja, osigurava ekvivalentnost konkurentnog izvršenja s nekim (unaprijed poznato kojim) serijskim izvršenjem, metoda vremenskih oznaka osigurava konkurentnost ekvivalentnog izvršenja sa specifičnim, unaprijed poznatim serijskim izvršenjem.

Ideja ove metode:

- Svakoj transakciji dodijeljuje se jedinstvena vremenska oznaka. Ona se dobije dopisivanjem rednog broja lokacije oznaci vremena aktiviranja transakcije (oznaka lokacije je neophodna jer se više transakcija na više lokacija može istovremeno aktivirati).
- Ažuriranja se fizčki upisuju u bazu tek pri uspješnom kompletiranju transakcije
- Svaki fizički primjerak logičkog objekta u bazi nosi vremensku oznaku transakcije koja ga je posljednja ažurirala (raznim optimizacijama vremenske oznake objekata u bazi mogu se izostaviti)
- Svaki zahtjev starije transakcije T1 za operacijom odbacuje se, a transakcija se poništava i ponovno aktivira, ukoliko je taj zahtjev u konfliktu sa operacijom koju je već, nad istim objektom izvršila mlađa transakcija T2. Operacija transakcije T1 je u konfliktu s operacijom tranakcije T2 ako se izvršavaju nad istim fizičkim objektom, i bar jedna od tih operacija je ažuriranje.
- Ponovno aktiviranoj transakciji dodijeljuje se nova vremenska oznaka

Oporavak

Da bi se dobila atomičnost transakcije u distribuiranom okruženju, u smislu da se izvrše sve njene radnje ili nijedna, sustav za ipravljanje transakcijama mora osigurati da se svi procesi te transakcije uspješno kompletiraju ili da se svi njihovi efekti ponište. Na primjer, da bi se izvršilo ažuriranje logičkog objekta x, od strane neke transakcije, potrebno je uspješno izvršiti ažuriranje svih primjeraka tog objekta. Taj zadatak zahtijeva posebno razmatranje zato što su primjerci jednog objekta na raznim lokacijama koje, nezavisno jedna od druge, mogu biti onemogućene da izvrše tu radnju (npr.zbog pada sustava). Da bi se osiguralo da ažuriranja svih primjeraka svih objekata od strane jedne transakcije budu uspješno izvršena, ili da nijedno ne bude izvršeno, primjenjuje se poseban protokol dvofaznog komplementiranja transakcije. Njemu se pristupa kada je koordinator transakcije obaviješten da su svi procesi te transakcije završili sa radom(uspješno ili neuspješno), i sastoji se od dvije faze:

- Za svaki logički objekt x koji transakcija T ažurira, i za svaki primjerak x_i objekta x, koordinator te transakcije šalje poruku lokaciji i (njenom SUBP) da preliminarno ažurira objekt x_i. Ako je proces transakcije T na lokaciji i us-

pješno izvršen, lokacija i odgovara tako što izvrši preliminarno ažuriranje, tj. prepisuje vrijednost x_i iz radnog prostora transakcije u svoju log datoteku, i obavještava o tome koordinatora; u suprotnom slučaju, lokacija obavještava koordinatora o neuspjehu.

- Ako je koordinator obaviješten da su sve radnje preliminarnog ažuriranja svih objekata koje transakcija T ažurira, na svim lokacijama, uspješno obavljene, on šalje poruke tim lokacijama da izvrše operaciju konačnog ažuriranja svih svojih primjeraka svih objekata koje T ažurira; lokacije odgovaraju tako što odgovarajuće vrijednosti iz svojih log datoteka prepisuju u svoje lokalne baze i oslobađaju resurse koje je transakcija T držala. Kada se obave sva ta prepisivanja i o tome obavijesti koordinator, izvršenje transakcije T je završeno. Ako je koordinator obaviješten da bar jedna lokacija nije uspješno obavila prvu fazu, on šalje poruku svim lokacijama o poništenju transakcije.

Ovakav protokol dvofaznog komplementiranja transakcije u implementacijama se poboljšava na razne načine, u cilju smanjenja broja poruka koje se prenose. Npr., moguće je pretpostaviti da će komplementiranje biti uspješno i time eliminirati slanje poruka iz prve faze. Ukoliko je kompletiranje zaista uspješno, broj poruka je znatno smanjen; ukoliko je kompletiranje (a pretpostavlja se da je to rijeđi slučaj), broj poruka se uvećava,a zahtijeva se i poništavanje određenog broja radnji.

Heterogeni distribuirani sustavi

Prerpostavka o homogenosri DSUBP, tj. pretpostavka da sve lokacije u DSUBP imaju isti SUBP pokazala se kao prejako ograničenje u današnjim uvjetima, kada velike količine podataka i aplikacija postoje na raznim računalima, pod različitim operatuvnim sustavima i pod kontrolom različitih SUBP. Potreba za istovremenim pristupom ovakvim podacima unutar jedne aplikacije, ili čak i jedne transakcije, postavlja zahtjev pred proizvođače DSUBP da osiguraju podršku heterogenim sustavima. To znači da, pored nezavisnosti pristupa podacima i obrade podataka od lokacije, fragmentacije, ponavljanja podataka, stroja, operatvinog sustava i mrežnog protokola, heterogeni distribuirani sustav mora osigurati i nezavisnost od SUBP na pojedinim lokacijama.

Postoje dva različita pristupa riješavanju ovog problema. Jedan je izgradnja tzv. sustava multibaza podataka, SMBP. SMBP je programski sustav koji se sastoji od niza komponenti. Jedna od njih je jedinstveni jezik za kreiranje podataka i manipuliranje podacima koji su pod kontrolom heterogenih SUBP.

Druga komponenta SMBP je globalni upravljač transakcija. SMBP osigurava, pored lokalnih transakcija (nad jednim SUBP), i upravljanje globalnim transakcijama. Globalne transakcije se sastoje od većeg broja podtransakcija koje se izvršavaju nad pojedinačnim (različitim) SUBP, i sa njihove pozicije se ponašaju kao lokalne transakcije.

SMBP uključuje i skupove servera, po jedan za svaki lokalni SUBP, koji se ponašaju kao veza između globalnog upravljača transakcija i lokalnog SUBP. Svaka globalna

transakcija predaje globalnom upravljaču transakcija operacije čitanja i upisa, a ovaj ih možr predati na obradu lokalnim SUBP (preko odgovarajućih servera), odložiti ili prekinuti transakciju. Kada globalni upravljač transakcija odluči kopletirati globalnu transakciju, on upućuje komandu za kompletiranje lokalnim SUBP. Globalni upravljač transakcija je odgovoran za funkcionalnost i za održavanje svojstava globalnih transakcija, (ACID svojstva, globalna lineariziranost izvršenja skupa transakcija izbjegavanje ili razriješavanje uzajamnog blokiranja, oporavak od sistemskih padova).

Oblast sustava multibaza je još uvijek otvorena istraživačka oblast. Osnovni cilj ovih sustava je da kroz upravljanje globalnim transakcijama održavaju konzistentnost multibaze.

Drugi pristup heterogenim sustavima je komercijalno zastupljeniji. Radi se o zgradnji aplikacijskih programa, tzv. prolaza (eng. gateway) sustava SUBP1 prema sustavu SUBP2 koji omogućava korisnuku SUBP1 (na lokaciji l_1) da komunicira sa SUBP2 (na lokaciji l_2).

Realizacija aplikacijskog programa prolaza sustava SUBP1 prema sustavu SUBP2 je u nadležnosti sustava SUBP1, a program se izvršava nad sustavom SUBP2. Npr., ako je SUBP1 sustav Oracle a SUBP2 sustav DB2, onda bi prolaz sustava Oracle prema sustavu DB2 omogućio korisniku sustava Oracle da komunicira sa sustavom DB2, „lažno“ predstavljajući sustav DB2 kao Oracle.

Aplikacijski program prolaz ostvaruje kroz:

- protokol za razmjenu informacija između SUBP1 i SUBP2
- relacijski server za SUBP2
- preslikavanja između tipova podataka i upitnih jezika dva sustava
- preslikavanje strukture kataloga sustava SUBP2 u strukturu kataloga sustava SUBP1
- učešće u dvofaznom protokolu kompletiranja transakcija
- doslijednu primjenu mehanizma zaključivanja, itd.

S obzirom na značaj koji ima realizacija kvalitetnih programa prolaza, postoji razvijena aktivnost standardizacije odgovarajućih protokola. U implementaciji pune funkcionalnosti prolaza javljaju se značajni problemi, pa zato komercijalni proizvodi ovog tipa ne podržavaju sve potrebne funkcije.