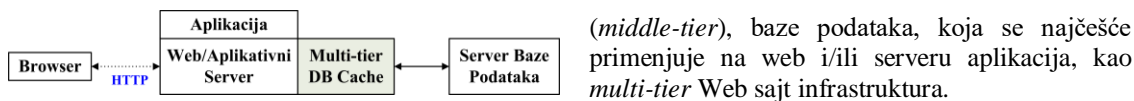


## Proof of Concept

**Cilj koncepta:** predstaviti arhitekturu aplikacije kako će funkcionisati u situaciji kad broj korisnika preraste jedan server.

Vreme odziva i pouzdanost su dva ključna elementa koje razdvajaju dobre od loših *World Wide Web* (Web) sajtova. Pojedina istraživanja pokazala su da sporo reagovanje Web sajtova prilikom preuzimanja sledeće web stranice može da pogubno utiče na Web sajt. Neke eksperimentalne procene pokazuju da 30 % korisnika napušta Web sajt ako je vreme preuzimanja sledeće web stranice duže od 8 sekundi, a stopa napuštanja dostiže 70 % kada je vreme za preuzimanje oko 12 sekundi. To pokazuje značaj brzog vremena odziva Web sajta da zadrži svoje klijente. Ovakve neželjene okolnosti, koji su posledica velikog broja korisnika, mogu se preduhitriti primenom različitih Internet zasnovanih tehnologija, metoda i tehnika. Primena različitih Internet baziranih arhitektura, tehnologija, metoda i tehnika u mnogome su uticale na skalabilnost primenjenih sistema koja je poboljšale performanse sistema kao posledica povećanog broja zainteresovanih korisnika kompanijskih Web sajtova.

**Data caching** - različite tehnike keširanja (*caching*) su primenjivane kao odgovor na sve veće zahteve Interneta za povećanje performansi web aplikacija. Takve primene, obično, postižu određene uspehe u poboljšanju mere skalabilnosti na aplikativnim i web serverima koji rade na više relativno jeftinijih računarskih sistema. Međutim, time se ne rešava problem skalabilnosti za servere baze podataka. Jedan od mogućih, rešenja ovog problema je primena srednjeg sloja



**Slika 1.** *Multi-tier* keširanje servera baze podataka

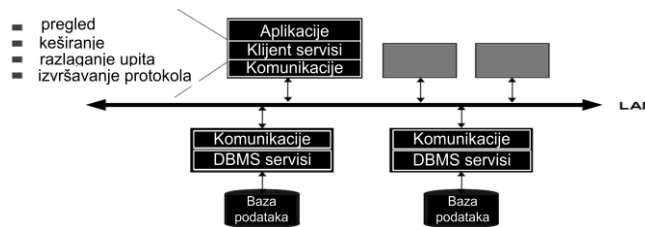
*Multi-tier* web sajt arhitektura, keširanje srednje nivoa baza podataka, omogućava sledeća poboljšanje sistema: (a) *Skalabilnost*: deljenje opterećenja upita jedinstvenom serveru baze podataka na više na više jeftinih računarskih sistema pomoću Web/Aplikativnih servera i *multi-tier DB Cashe* sistema; (b) *Dostupnost*: nastavak servisa za aplikacije koje zavise samo od keširanih tabela čak i ako je server baze podataka nedostupan; (c) *Performanse*: ujednačavanje vrhova opterećenja koje su posledica povećanog opterećenje celokupne infrastrukture sistema.

**Connection pooling** - Otvaranje i održavanje konekcije pojedinačnog internet korisnika i generisanje dinamičke *database-driven* Web sajt aplikacije korišćenjem baze podataka, predstavlja skupu aktivnost i nepotrebno trošenje celokupnih infrastrukturnih resursa. U cilju smanjivanja trošenja ovih resursa, nakon što je ostvorena konekcija, ona se smešta u „bazen” (*Connection pool*). Ponovo korišćenje konekcije, kada bude potrebna, preuzima se *Connection pool* bazena tako da se nova konekcija ne uspostavlja ponovo a uspostavlja se nova konekcija samo ako postojeća konekcija nije dostupna. Time se smanjuje vreme koje internet korisnik mora da čeka za uspostavljanje nove konekcije prema bazi podataka.

**Indeksiranje baze podataka** - Indeksiranjem baze podataka formira se struktura podataka koja poboljšava brzinu rada i pronalaženje podataka u tabeli baze podataka. Međutim, ovo unapređenje ima svoju „cenu” koja predstavlja dodatni upis i održavanje strukture indeksa podataka u skladišnom prostora servera baze podataka.

**Distribuirana baza podataka** - je kolekcija velikog broja, logički integrisanih baza podataka, distribuiranih preko određene heterogene računarske mreže. Sistem za upravljanje distribucionim bazama podataka je softver koji upravlja mehanizmom pristupa distribuiranim bazama podataka i čini ga transparentnim za sve korisnike. To je više ili manje spregnut multiprocesorski računarski sistem sa sinhronizovanom vremenskom podelom zadataka.

Za što efikasnije iskorišćavanje softverskih, hardverskih i komunikacionih resursa prilikom projektovanja distribuirane baze podataka neophodno je izvršiti pravilnu fregmentaciju, alokaciju i replikaciju fragmenata i optimizaciju baze podataka. U nekim specifičnim slučajevima potrebno je izvršiti optimizaciju postavljanja korisničkih upita. *Replikacija* znači da se na određenim lokacijama čuva više kopija baze podataka, relacija, baze podataka ili relacionih fragmenata.



**Slika 2.** Distribuirana baze podataka i komunikacije

Za što efikasnije iskorišćavanje softverskih, hardverskih i komunikacionih resursa prilikom projektovanja distribuirane baze podataka neophodno je izvršiti pravilnu fregmentaciju, alokaciju i replikaciju fragmenata i optimizaciju baze podataka. U nekim specifičnim slučajevima potrebno je izvršiti optimizaciju postavljanja korisničkih upita.

**Computer Clusters** – je arhitektura računara koji su raspoređeni na takav način da podržava drugačiju svrhu od klasičnih poslovnih sistema opšte namene. Njihova arhitektura je prvenstveno namenjena: (1) kao što je podrška web servisa; (2) intenzivnom računanju naučnih proračuna. U osnovi postoje tri vrste klastera, to su:

- **Load-Balancing Cluster** – Klaster (*Cluster*) zahteva efikasnu mogućnost za balansiranje rad među dostupnim računarima. U ovoj arhitekturi, klaster čvorovi dele računarsko opterećenje za poboljšanje ukupne performanse sistema. Klaster visokih performansi koji se koristi za naučne proračune će balansirati rad različitih algoritama. S druge strane, Web-server klaster može da bude dodeljen bilo kom drugom nodu koristi se na farmama web servera (*Web farma*).
- **High-Availability Clusters** – Oni nude veliku verovatnoću da će svi resursi biti u potpunoj funkciji. Ako se kvar dogodi, kao što slučaj kada „sistem padne” onda su izgubljeni svi upiti koji su bili u toku. Bilo koji izgubljen upit, ako bude ponovo zahtevan biće obrađen na drugom server računaru u grupi. Ovaj tip klastera se široko koristi kao web-server, *e-mail* server, *news*-server ili *ftp*-server.

Prednosti korišćenja klaster kompjuter su mnogobrojne a najvažnije su:

- **Apsolutna skalabilnost** – Moguće stvoriti velike serverske klastere koji imaju veću moć (procesorsku, memorijsku) čak i od najvećih samostalnih računara. klaster može imati desetine multiprocesorskih mašina.
- **Dodatna skalabilnost** – Klaster je konfigurisan tako da je moguće da dodaju novi sistemi uz minimalnu nadogradnju. Klasteri imaju mogućnost da dodaju sistema horizontalno proširenje, a to znači da više računara mogu da se dodaju klasteru a da pri tome klaster poboljša svoje performanse, redundantnost ili toleranciju greške.
- **Visoka dostupnost** – Svaki čvor u klasteru je samostalan kompjuter, neuspeh jednog čvora ne znači gubitak servisa. Jedan čvor može da se isključi iz postojeće arhitekture i odnese na održavanje, dok ostatali deo klastera preuzima zadatke tog pojedinačnog čvora.
- **Prihvatljiv odnos cena/performance** – Klasteri su obično podešeni da poboljšaju performanse i dostupnost u odnosu na pojedinačni računar, a obično je mnogo isplativiji od pojedinačnih računara u odnosu na istu brzinu i dostupnost.

**Client-Side Web Caching** – Tipično „keš” se nalaze u pretraživačima i u bilo kojoj web sredini između korisnika i servera. Web keš je podeljen na „kratkoročni keš” koji prima web objekte sa Interneta direktno, i dugoročne memorije koja prima web objekte iz kratkoročnog keša. Kada se korisnik usmerava ka određenoj web stranici, svi web objekti ugrađeni u stranice, pre svega, se čuvaju u kratkoročnom kešu. Web objekti koji su posećeni više puta biće izmešteni u dugoročnu memoriju za duže keširanje, ali sa druge strane drugi najstariji objekti će biti uklonjeni iz dugoročnog keša. *Client-Side Web Caching* obezbeđuje da su poželjni web objekti sačuvani što duže vreme, dok su manje korišćeni objekti ranije uklonjeni i time obezbedi maksimalni „hit” odnos.