

2. Osnovni zahtjevi za posredničke sustave

POUZDANOST

- Sposobnost sustava da obavlja svoje zadaće besprekidno unutar određenog vremenskog perioda
- Postiže se modelima:
 - o uvišestručavanja i konzistentnosti
 - doprinosi pouzdanosti
 - problem je održavanje replika konzistentnima
 - stroga konzistentnost, slijedna konzistentnost, povezana konzistentnost, konzistentnost redoslijeda, slaba konzistentnost, konzistentnost prilikom otpuštanja, konzistentnost prilikom zauzimanja
 - o otpornošću na greške
 - omogućava raspodijeljenom sustavu da otkriva, dojavljuje i uklanja pogreške pri svom radu
 - privremene (pojave se jednom i nestanu), povremene (pojavljuju se i nestaju), trajne (prisutne dok se ne otkriju i uklone)

RAZMJERAN RAST

- Sposobnost razmjerne prilagodbe: veličini, prostornoj rasprostranjenosti, načinu upravljanja
- Ustupak svojstva razmjernog rasta i radnih svojstava (engl. scalability/performance trade-off) (vrijeme kašnjenja, mrežni promet, radno opterećenje)
- Za održanje radnih svojstava sustava uz povećanje broja korisnika uz prihvatljive troškove potrebno je: uvišestručavanje sredstava, prostorna raspodijeljenost sredstava, asinkrona komunikacija dijelova sustava

SIGURNOST

- Teško ostvariti dobar model sigurnosti zbog raširenosti na sve komponente
- Uobičajene sigurnosne prijetnje: presretanje, prekid, izmjena, umetanje
- Sigurnosni mehanizmi: enkripcija, autentifikacija, autorizacija, praćenje rada

KVALITETA USLUGE

- Quality-of-Service (QoS)
- Skup objektivnih parametara oko kojih se i korisnici i pružatelji usluga slažu da u nekoj mjeri opisuju kvalitetu usluge (mrežna propusnost, brzina prijenosa podataka, izgubljeni paketi pri komunikaciji, prioriteti podataka/korisnika/aplikacija, prioriteti očuvanja podataka, vrijeme isteka sredstva)

KVALITETA DOŽIVLJAJA

- Quality-of-Experience (QoE)
- Definira korisnikov subjektivni doživljaj kvalitete usluge
- Najčešći faktori koji utječu na QoE: cijena korištenja, pouzdanost sustava, efikasnost sustava, sigurnost i privatnost sustava, korisniku bliska sučelja, korisnikov osjećaj sigurnosti u sebe pri korištenju

TRANSPARENTNOST

- Prikriva činjenicu da su sredstva raspodijeljena
- Razine: pristup, lokacija, migracija, premještanje, uvišestručavanje, paralelnost, kvar

OTVORENOST

- Pruža usluge sukladno normiranim pravilima te definiranoj sintaksi i semantici
- Otvoreni standard je specifikacija koja je: javno dostupna svima, stvorena i razvijana demokratskim putem (nije nečije vlasništvo), nije komercijalna (proizvodi jesu), neovisna o tehnologiji
- Pretpostavka za: međudjelovanje, prenosivost, proširljivost

MEĐUDJELOVANJE (interoperability)

- Postojanje komponenti različitih proizvođača koje mogu zajedno raditi
- Međusobno djelovanje različitih uređaja

PRENOSIVOST (portability)

- Omogućiti izvođenje aplikacija na različitim sustavima

PROŠIRLJIVOST (extensibility)

- Mogućnost dodavanja novih usluga i sredstava
- Potrebno imati dobro definiran i standardiziran skup sučelja (API) te modularnost sustava

PRILAGODLJIVOST (adaptability)

- Mora poznavati svoje komponente i njihov status
- Dinamički podesiv
- Pronalazi optimalno ponašanje za postizanje cilja
- Optimalno koristi sredstva

3. Scalability

Dimensions of scalability:

- System size (number of machines)
- Application size (machine load and communication traffic)
- Geographic distribution
- Information, computing, and communication technology (hardware, software, protocols, languages, methods, ...)
- Security and privacy
- Social and legal dimension
- Manageability

Potential bottlenecks and problems

- Computational and communication complexity of application algorithm
 - Centralized algorithms
- Network infrastructure
- Architecture
- Communication, collaboration, and synchronization
 - Traffic and node load
 - Synchronous communication
 - Pushing information, server initiated communication
- Data storage and management
 - Centralized data storage, linear lists, single file tree
- Strict semantics, consistency, and coherence

How to deal with scalability problem

- Decentralized algorithms
 - System partition into the smaller independent units
 - No machine has complete information about the system
 - Machines make decisions based only on local information
 - Failures of one machine does not ruin the algorithm
 - There is no implicit assumptions that a global clock exists
- Data storage and management
- Communication, collaboration, and synchronization
 - Asynchronous communication
 - Pulling information, client initiated communication
- Weaker guarantees for semantics, consistency, and coherence
- Limited stateful solutions

Large scalability

Limited growth

- Physical limiters
 - Signal propagation and power dissipation

- Bus-based multicomputer systems
 - 25-100 nodes
- Hardware architectures limiters
 - Cross-sectional bandwidth
 - Bus-based multiprocessors
 - Up to 32 processors
 - Multiprocessors based on hierarchy of rings
 - Up to 100 processors
- Software architectures limiters
 - Communication and information management protocols
 - Multicomputer systems
 - Up to 250 nodes

Intranet systems

- Clusters
 - Homogenous systems
 - Ultra-high-performance, special-purpose interconnection networks
 - High degree of centralized control
 - Computational model
 - Synchronous communication
 - Distributed shared memory
 - Message passing
 - Programming
 - Resource allocation and processes management
- Local-area networks
 - Heterogeneous systems
 - High reliable communication based on broadcast
 - Geographical distribution
 - Separate administration
 - Lack of global knowledge
 - Limited centralized control
 - Computational model
 - Loosely synchronous communication, RPC
 - CORBA, Java RMI, DCOM
 - Client/server
 - Programming
 - Connection to established services that encapsulate hardware resources or provide defined computational services

Design principles

- Trade-offs of
 - Performance
 - Security and privacy
 - Usability
 - Functionality
- Performance
 - Latency, traffic, and workload
 - Coherence and consistency

- Energy dissipation

Design example

- AT&T Labs, IP Technology Organization
 - 1995 – 2000
 - Middletown, NJ, San Mateo, San Jose, CA
- GeoPlex Platform
 - The common open IP platform is a collection of reusable software components creating a framework for deploying secure, authenticated IP services over the open Internet or in internal intranets

GeoPlex distributed cache manager

Motivation

- Ambiguity of definition of the coherence
- Lack of coherence protocol
- Performance
- $T_{Avr} = r_{Hit} \times t_{Hit} + (1 - r_{Hit}) \times t_{Miss}$
- To reduce the time parameters
- To increase the hit rate
- Harvest cache

Experience

- To increase the hit rate
 - Connecting the larger number of clients to the same proxy machine increases the hit rate
 - Increasing the number of clients increases the probability that they are interested in the same data object
- Proxy load and communication limited capabilities
 - Proxy machine that runs the cache limits the number of the clients connected to the same machine

Solution

- Distributed cache
 - To enable the proxy machines to communicate
 - The communicating proxy machines act as a single distributed cache
- Scalable distributed cache manager
 - In order to increase the hit-rate, the number of clients per one distributed cache should be increased
 - The main design issue is the scalability of distributed cache manager

Performance comparison

Simple analytical model improves the performance of the system in the early phase of design and guides the implementation of scalable system.

Worldwide scalability

Unlimited growth

- Autonomous and independent domains
 - Separately managed and administrated
 - Distribution and hierarchy
- Interaction and communication
 - Ad hoc and spontaneous
 - Message oriented, document based
- Computational model
 - Collaboration and competition
 - Data mining
- Programming
 - Brokering, negotiation, and trading

Internet systems

Wide-area networks, internetworked systems

- Worldwide distribution
- Unreliable, point-to-point communication
- Lack of centralized control
- International issues
- Communication
 - Asynchronous communication
- Program execution
 - Different mobile code models
 - Remote control

Design principles

Aggregation

- Individual entities of a given type owned by one domain are aggregated and exported as a single unique entity
- It reduces the amount of information about a given domain that is exported to other domains
- Both time and space efficient and scalable

Lazy evaluation

- Actions are only partially (“lazy”) evaluate by one domain
- Partial evaluation uses as input parameters only the entities of the given domain
- The results of the partial evaluation are sent to the another domain, where the rest of the evaluation is done by using entities from that domain
- Only space efficient and scalable, but it could be time consuming
- The execution of the action could be spread out through multiple domains, and time for execution could be long

Replication based on caching

- In order to improve the performance of the lazy evaluated functions, some of the values of the entities are replicated from one to the another domain
- Replication is done only by request (Lazy replication, or caching)
- This means that value of the entity is replicated only if action that is executed need this value
- Since the copy of the value must be coherent with the value of the entity in the originating domain, coherence protocol should be introduced
- Protocol maintains copies of the same value coherent
- The basic features of the coherence protocol are weak coherence and coherence window

GeoPlex multiple clouds architecture

Clouds

- Autonomous and independent domains
- Constitutes an authentication trust and a single registration domain
- Centralizes authentication, access control and security
- Cloud registers, authenticates, and authorized users, services and other clouds

Domain Name System (DNS)

Directory service

- Keep track of locations of resources
- Provide people-friendly names for resources

Cell organization

- Cell directory server (CDS)
 - Stores the names and properties of the cell’s resources
- Replicated and distributed database system
- Worldwide scalable
- Unique resource name
 - Name of cell followed by name used within cell
- Resource location mechanisms
 - GDS - Global Directory Service (It uses X.500 standard)
 - DNS- Domain Name Server (It uses Internet naming system)
 - ONS - Object Name System (EPC – Electronic Product Code, RFID)

Domain name system

- Organized machines in cells – domains
- Mapping
 - Mapping of host names and e-mail destinations to IP addresses
- Generalized database system
 - Distributed, hierarchical, and worldwide scalable
 - Stores variety of information relating to naming

4. Posrednici zgodom oblikovanih mreža

- engl. Ad-hoc networks
- Svaki čvor mreže je mogući usmjernik
- Decentralizirana mreža
- Promjenjiva topologija mreže

POKRETNE ZGODOM OBLIKOVANE MREŽE

- Mobile Adhoc networks (MANET)
- Čvorovi mreže su pokretni uređaji
- Promjena položaja uređaja utječe na karakteristike mreže
- Ograničeni resursi uređaja
- prednosti: mreža bez infrastrukture (primjena: vojska, službe spašavanja, poslovni sastanci,...)
- nedostaci: česte promjene topologije, ograničenost resursa, složena izrada raspodijeljenih primjenskih sustava

ZAHTJEVI MANET POSREDNIČKIH SUSTAVA:

- **Heterogenost**
 - Omogućiti spajanje raznolikih uređaja na mrežu
 - Programske apstrakcije za povezivanje sustava zasnovanih na različitim mrežnim arhitekturama
- **Pokretljivost i promjenjiva topologija**
 - Promjena položaja uređaja utječe na topologiju
 - Česte i nepredvidljive promjene
 - Dinamično uspostavljanje veza među čvorovima
 - Prilagodba topologije s obzirom na dostupnost alternativnih mreža
- **razmjern rast**
 - dodavanje čvorova mreže bez smanjenja performansi, pozicija novog čvora ne narušava rad mreže!
- **upravljanje resursima**
 - stanje baterije, sposobnost obrade podataka, memorijski prostor
 - povezivost okoline, propusnost mreže
- **kvaliteta usluge**
 - osigurati propusnost i pouzdanost mreže

- **sigurnost**
 - nesigurni usmjernici(čvorovi), česte promjene puta podataka zbog promjene topologije
- **ujednačavanje sudjelovanja**
 - glavna uloga čvora nije usmjeravanje podataka
 - zahtjevi za štedljivim upravljanjem resursima oprečni usmjeravanju nekorisnih podataka
 - pohlepni čvorovi narušavaju rad mreže!

UPRAVLJANJE POTROŠNJOM ENERGIJE

- **centralizirano upravljanje**
 - jedan čvor zadužen za prikupljanje podataka o energetsom stanju mreže
 - optimalno prosljeđivanje podataka s obzirom na potrošnju energije
 - nedostatak: potrebna veza s glavnim čvorom
- **raspodijeljeno upravljanje**
 - čvor prikuplja podatke iz svoje okoline i samostalno upravlja potrošnjom energije
 - nedostatak: lokalno energetska stanje mreže

PROTOKOLI USMJERAVANJA PODATAKA U MREŽI

- **Jednorazinski**
 - zasnovani na tablici (OLRS, DSDV, WRP)
 - princip preuzet iz žičanih mreža
 - periodička razmjena tablica s podacima o svim čvorovima
 - čvorovi u svakom trenutku imaju približnu sliku mreže
 - brzo usmjeravanje podataka, ali potrebna razmjena velike količine podataka prilikom osvježavanja tablice
 - pogodno za mreže s rjeđim promjenama u topologiji
 - reaktivni (DSR, AODV)
 - osvježavanje topoloških podataka po potrebi
 - put podataka određuje se prije prosljeđivanja, šalje se broadcast poruka za otkrivanje puta
 - dobre performanse u prosjeku, moguće veliko kašnjenje prilikom broadcasta(zagušenje mreže)
 - hibridni (ZRP, TORA)
 - kombinacija tabličnih i reaktivnih
 - mreža podjeljena na podmreže
 - usmjeravanje podataka unutar podmreže pomoću tablica, a izvan podmreže reaktivnim algoritmima
- **Hijerarhijski (CBRP, CEDAR)**
 - Poopćenje hibridnih protokola
 - Mreža se rastavlja na hijerarhijske cjeline
 - Tablični ili reaktivnih algoritmi za pojedinu cjelinu unutar hijerarhije
 - Odabire se čvor unutar hijerarhijske razine za osvježavanje topoloških podataka
 - Neprikladni za mreže uređaja s malo resursa

- **Geografski (GPSR, BGR)**
 - Zasnovani na položaju čvorova unutar mreže
 - Čvor osvježava podatke o svojim susjedima s kojima je izravno povezan
 - Podaci se proslijeđuju onom susjedu koji je najbliži čvoru
 - Dobro podržana pokretnost čvorova

AODV PROTOKOL

1. postoji valjani zapis u tablici usmjeravanja: podaci se proslijeđuju prema odredišnom čvoru
2. ne postoji zapis u tablici proslijeđivanja: šalje se zahtjev za otkrivanje puta svim susjednim čvorovima (broadcast)
3. ponavlja se 2. korak dok zahtjev ne stigne do čvora sa zapisom u tablici usmjeravanja ili do odredišnog čvora
4. odgovor se šalje na adresu izvorišnog čvora (ne koristi se broadcast)

PREVENTIVNO USMJERAVANJE

- nadogradnja reaktivnih algoritama usmjeravanja
- promjena puta podataka prije prekida veze
- prekid veze uzrokuje dodatnu komunikaciju (ponovna slanja)
- otkrivanje kvalitete veze (snaga signala, starost zapisa u tablici, udaljenost, kolizije)
- Širina preventivnog područja može biti:
 - Prevelika – nepotrebni zahtjevi za promjenom puta
 - Premalena - nedovoljno vremena za uspostavu novog puta
 - $T_{\text{upozorenje}}$ (vrijeme proteklo od upozorenja do prekida) = T_{oporavak} (vrijeme potrebno za uspostavu novog puta)

SUSTAV ZA UJEDNAČAVANJE SUDJELOVANJA

- Jednostavan sustav kažnjavanja
 - Faktor sudjelovanja $U=f(U_{\text{poslani_paketi}}, U_{\text{primljeni_paketi}}, U_{\text{utrošena_energija}})$
 - Smanjivanje faktora sudjelovanja čvorovi kažnjavaju smanjujući svoj faktor sudjelovanja -> smanjivanje ukupne propusnosti mreže
 - Nedostaci: određivanje propusnosti na razini mreže, neotpornost na kvar čvora i zlonamjerne čvorove
- Sustav zasnovan na ugledu
 - Čvorovi održavaju tablicu ugleda (povijest proslijeđivanja)
 - Proslijeđivanje/odbijanje proslijeđivanja = veći ugled/manji ugled čvora
 - Zahtjev za proslijeđivanjem od čvora s „negativnim“ ugledom neće biti obrađen
- Sustav zasnovan na cijenama
 - Postoji središnji sustav za nadgledavanje sudjelovanja – blagajna
 - Čvorovi koji proslijeđuju podatke zarađuju „novac“ iz blagajne
 - Čvorovi izvorišta prometa uplaćuju novac u blagajnu

Posrednici senzorskih mreža

- Wireless Sensor Networks (WSN) – skup senzorskih uređaja povezanih wirelessom
- mreža od velikog broja čvorova s ograničenim resursima – $V_{\text{sensor}} \geq 1\text{mm}^3$
- poveznica fizičkog i virtualnog svijeta
- široka primjena: mjerenje seizmičke aktivnosti, praćenje divljih životinja, praćenje prometa, vojni sustavi, ...

VRSTE SENZORSKIH MREŽA

- upravljanje,
- samoupravljanje,
- hibridne

PREGLED ZAHTJEVA POSREDNIKA SENZORSKIH MREŽA

- **Upravljanje resursima**
 - Inherentno ograničeni resursi(energija, proc snaga, memorija, propusnost)
 - Prilagodljivo upravljanje resursima s obzirom na potrebe primjenskog sustava
 - Uspostavlja optimalnog odnosa između kvalitete rada i potrošnje resursa
- **Upravljanje topologijom**
 - Zgodom oblikovane mreže
- **Održavanje**
 - Mehanizmi za samostalno početno postavljanje i održavanje
- **Skupljanje i obrada podataka**
 - Središnja uloga senzorskih mreža
 - Obrada podataka je raspodijeljena, agregacija podataka
- **Pružanje programskih apstrakcija**
 - Izlaganje senzorskih mreža kao objekata
- **Podrška za OS**
 - Funkcionalno razdvajanje sučelja OS-a i posredničkog sustava

VRSTE POSREDNIKA SENZORSKIH MREŽA

- Zasnovani na bazi podataka
 - Pristup mrežama kao raspodijeljenoj bazi podataka
 - Upiti slični SQL-u prošireni vremenskim naredbama
- Zasnovani na pokretnim agentima
 - U mrežu se umeće primjenski sustav u obliku pokretnog agenta koji obilazi čvorove i prikuplja podatke (agregaciju podataka obavlja agent)
- Zasnovani na događajima
 - Posrednički primjesni sustav postavlja događaje u čvorovima mreže
 - Događajima se opisuje stanje senzora koje je potrebno dojaviti posredničkom sustavu
 - Asinkrona komunikacija

Posrednici za RFID(radio frequency identification) mreže

- Otkrivanje prisutnosti objekta
- RFID mreža može biti čitač ili transponder
- Primjena: skladišta, proizvodni pogoni, pametna kuća....

VRSTE POSREDNIKA RFID MREŽA

- aktivne(vlastito napajanje, adhoc, senzorske...)
- pasivne(napajanje iz signala čitača)

PREGLED ZAHTJEVA ZA POSREDNIKE RFID MREŽA

- **Filtriranje podataka**
 - Ograničena propusnost mreže
 - Usmjeravanje sadržaja
 - Mehanizam pretplate(primjenski sustav se pretplaćuje na podatke i RFID čitači obrađuju upite za kojima postoji interes)
- **Agregacija podataka**
 - Objedinjavanje više povezanih događaja otkrivanja(rasterećenje mrežnog prometa)
 - Izgladivanje pogrešnih očitavanja
- **Pristup memoriji transpodera**
 - Osigurati transparentan pristup memoriji transpodera i zalihosti podataka
- **Pouzdanost**
 - Ostvarivanje suradnje čitača s ciljem smanjenja lašnih očitavanja
- **Upravljanje čitačima**
 - Udaljenos postavljanje načina rada čitača
 - Osvježavanje programske podrške
- **Privatnost**
 - RFID: "kontroverzna tehnologija"
 - "Orwellovski svijet"

5.Quality of Service in Middleware

Systems

Key features od software quality:

- Correctness
- Completeness
- Scalability
- Fault-tolerance
- Extensibility
- Maintainability
- Documentation

Middleware classification

- Resource management
 - Software level resources (database middleware)
 - Hardware level resources (Infrastructure middleware)
- Application management
 - Application development and collaboration (application middleware)
 - Interapplication communication (communication middleware)

Quality assurance in context

- In engineering process: design and implementation (code quality), test (functional quality), optimization (nonfunctional quality)
- In deployment and execution: execution (quality of service (QoS) and quality of experience (QoE))

QoS – defines the functional and non functional characteristics of a computing system that are delivered as a service to end-user

- Hardware level QoS – koliko ima baterije, snaga procesora i sl.
- Connectivity level QoS – povezanost, da svi dobiju isti nivo usluge
- Data level QoS – što dobiju korisnici usluge (podatci), ovisi o web protokolima i njihovoj učinkovitosti. Korisnici podatke dobijaju pomoću: data caching, data replication (ak mu fali dio podataka onda iz spremjenih podataka napravi repliku tog dijela, neprecizno je)
- Application level QoS

QoE – user's subjective impression on the quality. GOMS:

- Goals – što korisniku treba
- Operations – operacije koje korisnik može izvršiti
- Methods – slijed operacija koje je potrebno učiniti da bi se ostvario cilj
- Selection rules – kriterij koji je korisnik izabrao za ocijeniti aplikaciju (vrijeme potrebno za napraviti i sl.)

Treba naći omjer između QoS i QoE. Bolji QoS (računalo -> objektivna mjera) često znači slabiji QoE (korisnici -> subjektivna)

Service level agreements - A formal contract which defines the terms under which service provider and consumer engage in interaction with the purpose of delivering and consuming a service.

Standardi:

- Web Service Level Agreement Language (WSLA)
 - SLA Contract Document Specification
- WSLA Service Deployment Information (WSLA SDI)
 - SLA Enforcement Policies
- WS-MetadataExchange
 - SLA negotiation protocol

6.Posrednici za pristup podacima i izvođenje transakcija

Transakcije – pristup za modeliranje i izgradnju sustava, pouzdani, otporni na greške

OSNOVNA ZAMISAO TRANSAKCIJE

- Preoblikuje sustav iz jednog konzistentnog stanja u drugo
- Niz operacija do konačnog konzistentnog stanja
- U međukoracima sustav može biti u nekonzistentnim stanjima

PRIMJER TRANSAKCIJE

- Prijenos srestava s jednog na drugi bankovni račun
- Dvije logičke operacije
- Skidanje/dodavanje sredstava s/na račun
- Ne smije doći do nekonzistentnog stanja(jedna od dvije operacije se ne izvrši)

ULOGA TRANSAKCIJSKIH POSREDNIKA

- Olakšati izgradnju i postavljanje pouzdanih transakcijskih primjenskih sustava koji podržavaju razmjerni rast
- Skriva se i izbjegava upravljanje transakcijama na niskoj razini, fokus na poslovnoj logici!!!

ACID MODEL TRANSAKCIJA

- **Atomarnost** – nedjeljiva operacija
 - o Transakcija uspjela – efekt transakcije se reflektira na stanje sustava
 - o Transakcija nije uspjela – operacije transakcije se poništavaju a sustav se vraća u inicijalno konzistentno stanje u kojem je bio prije početka izvođenja sporne transakcije
 - o Atomarnost se postiže DO-UNDO-REDO protokolima (2PC protokol)
 - 2PC, two phase protocol
 - U sustavu postoje: sudionici(upravljaju sredstvima), koordinator(upravlja transakcijama)
 - Prva faza: koordinator traži od svih sudionika u sustavu dopuštenje za potvrdu transakcije
 - Druga faza: nakon što svi sudionici dopuste potvrdu transakcije, koordinator potvrđuje transakciju
 - Sve operacije ostvarene kao logičke funkcije
 - Sve funkcije se zapisuju u dnevnike transakcije
 - DO(izvođenje transakcije)
 - UNDO(poništanje transakcije)
 - REDO(obnavljanje transakcije)
- **Konzistentnost** – sustav ide iz jednog konzistentnog stanja u drugo konzistentno stanje
- **Izolacija** – druge transakcije koje se možda izvode paralelno nemaju utjecaj na krajnji ishod
 - o Višeprocorski sustavi

- Razine izolacije
 - Serializable
 - Repeatable Read
 - Read Committed
 - Read Uncommitted
- Problemi
 - Dirty reads
 - Nonrepeatable reads
 - Phantoms
- 2LP, two phase locking protocol
 - Shared lock(S) za čitanje
 - Exclusive lock(X) za pisanje
 - Faza 1: transakcija mora zatražiti i dobiti sve potrebne ključeve za izvođenje svojih operacija
 - Faza 2: nakon što se izvrše sve operacije transakcije, transakcija otpušta sve ključeve
 - S ključ je u konfliktu s X ključem, X također s X
- **Trajnost** – jedna potvrđena transakcija ostaje trajna čak i u slučaju rušenja sustava
 - DO-UNDO-REDO dnevnici
 - U slučaju kvara:
 - REDO svih potvrđenih transakcija
 - UNDO svih otkazanih
 - DO svih nezavršenih

RASPODIJELJENJE TRANSAKCIJE

- Što ako se sruši pojedini lokalni sudionik ili koordinatorski?
 - Donose se heurističke, ad-hoc odluke
 - Može doći do narušavanja atomarnosti i konzistencije, ponekad je potrebna intervencija čovjeka da se sustav dovede u ispravno stanje

ACID VARIJACIJE

- Optimističan nadzor istovremenog izvođenja transakcija
 - Svaka transakcija dobije svoje kopije podataka
 - Tijekom potvrđivanja se razriješavaju konflikti ukoliko do njih dođe
- Ugnježdene transakcije
 - Podtransakcije
 - Dobiva ključ ako je slobodan ili ga ima roditelj
 - Kad se potvrdi podtransakcija, roditelj dobiva ključeve koje je ona držala
 - Kad se otkáže podtransakcija, ključevi se oslobađaju

OBJEKTNO ORIJENTIRANI TRANSAKCIJSKI POSREDNICI

- Razvoj iz RPC-a, uvodi se OO pristup u izgradnji raspodijeljenih sustava
- Eksplicitan API za transakcije, ključne riječi za početak i završetak transakcije(begin, commit, rollback)
- Enterprise JavaBeans, standard za izgradnju poslovnih aplikacija
- Poslužiteljski kod isti izazovi i zahtjevi
 - o Postojanost podataka
 - o Integritet transakcija
 - o Istodobno izvođenje
 - o Sigurnost
- Fokus na logici samog primjenskog sustava, EJB se brine o navedenim zahtjevima
- Opisnik za postavljanje , XML dokument koji sadrži sljedeće informacije
 - o Ime home sučelja
 - o Java razred za Bean(objekt koji drži logiku)
 - o Java sučelje za Home sučelje
 - o Sigurnosne postavke i prava pristupa
 - o Transakcijske postavke(razina izolacije, kontekst transakcije)

TRANSAKCIJE U OKOLINAMA ZASNOVANIM NA PORUKAMA

- RPC nedostaci
 - o Klijent i poslužitelj dostupni istodobno
 - o Klijent blokiran dok ne dobije odgovor od poslužitelja
 - o Čvrsto povezan sustav, nije uvijek ostvarivo
 - o Klijent možda želi obraditi zahtjev i ako poslužitelj odgovara
 - o Klijent možda želi poslati zahtjevi grupi poslužitelja a ne samo jednom
- Message-oriented middleware MOM (Posrednici zasnovani na porukama)
 - o MOM je posrednik između aplikacija
 - o Aplikacije komuniciraju međusobno preko MOM
 - o MOM pohranjuje poruke i odgovore
 - o MOM jamči isporuku zahtjeva ili odgovora
 - o Asinkroni komunikacijski model
 - o Fire and forget a message
 - o Aplikacije ne moraju biti dostupne u istom trenu
 - o Aplikacija nije blokirana i ne čeka ni u jednom trenu
- Modeli komunikacije porukama
 - o Point to point(red poruka)
 - Komunikacija preko Redova poruka
 - Posrednik između aplikacija koje komuniciraju
 - Red poruka je postojan spremnik za poruke
 - Red poruka ima svoj logički identifikator i njime upravlja Upravitelj Reda poruka (Queue Manager)
 - Redovi poruka se mogu koristiti kao transakcijski resursi, tada svaka operacija pisanja u ili čitanja iz Reda poruka ovisi o sudbini transakcije unutar koje se izvodi
 - o Publish/subscribe
 - Proizvođači poruke objavljuju na određene teme

- Potrošači se pretplate na određene teme i dobijaju sve poruke koje se objave na te teme
- Na ovaj način poruku može dobiti više različitih potrošača za razliku komunikacije preko Reda poruka

PROGRAMSKI MODELI

- J2EE Java Message Service, standardan API
 - uvodi se transcated session
 - Sve poruke unutar transakcijske sjednice postaju dio transakcije
 - Poruke se šalju i primaju jedino ako se transakcija potvrdi
- Message-driven Bean (MDB)
 - Integracija JMS i EJB
 - MDB je Bean u Containeru i pridružen je nekom Redu poruka ili nekoj Temi
 - U MDB opismiku je definirana metoda koju je potrebno pozvati i predati joj pristiglu poruku
 - EJB presreće JMS poruke i poziva odgovarajuće metode

TRANSAKCIJE NA WEBU

- Web Services tehnologija
 - Raspodijeljeni programski model
 - Arhitektura zasnovana na uslugama (SOA)
 - Standardi WSDL, UDDI i SOAP
- Potreba za pouzdanim transakcijama i komunikacijom
 - WS-Coordination
 - Opisuje radni okvir za koordinaciju Web transakcija
 - Stvaranje konteksta transakcije
 - Razmjenu konteksta među uslugama sudionicima
 - WS-Transaction
 - Atomic Transaction (AT)
 - Kraće transakcije
 - Unutar povjerljive domene
 - Bussines Activity (BA)
 - Duge transakcije
 - Unutar različitih domena
 - WS-ReliableMessaging
- Service oriented middleware (SOM)
 - Upravlja raspodijeljenim, decentraliziranim resursima na Web-u
- Bussines Process Execution Language for Web Services (BPEL)
 - Jezik za definiciju, koordinaciju i izvođenje poslovnih procesa izgrađenih od Web usluga
 - Tok podataka (Data Flow)
 - Tok upravljanja (Control Flow)
 - Definicija i izvođenje poslovnih transakcija
 - Rukovanje iznimkama

NAPREDNE TRANSAKCIJE

- **Long Running Unit of Work (LRUOW)**

- Model omogućuje istodobno izvođenje dužih transakcija bez zaključavanja pojedinih sredstava
 - Packaging Control – Poslovne aktivnosti grupira u Jedinice posla (Unit of Work, UoW)
 - Visibility Control – Pojedini objekti koji su stvoreni ili mijenjani su vidljivi samo unutar određenog konteksta
 - Concurrency Control – omogućuje se procesima da pristupaju istim podacima
- LRBP se modelira kao aciklički usmjereni graf čiji su čvorovi Jedinice posla (UoW)
- Svaki čvor ima jednog roditelja i može imati više djece
- Svaki podzadatak u poslovnom procesu se vezuje na jedan čvor u grafu i izvodi se u tom kontekstu
- Moguće je istodobno izvođenje nad istim podacima, pritom svaki proces dobija svoju kopiju podataka i objekata, te se sve bilježi u čvoru roditelju
- Jedinica posla se potvrđuje ili opoziva izvođenjem odgovarajuće metode nad čvorom
- Svaki pojedini UoW se izvodi u dvije faze
 - Long-running phase (izvođenje samog posla)
 - Short-running phase (razrješuju se posljedice istodobnog izvođenja)

- **Conditional Messaging (Komunikacija porukama uz uvjet)**

- Nadogradnja na standardni MOM
- Omogućuje definiciju različitih uvjeta kao nezavisnih objekata o kojima ovisi isporuka i obrada poruka
- Omogućuje nadzor isporuke ili obrade poruke primatelju slanjem poruke o potvrdi primitka ili završetka obrade
- Omogućuje izvrjednjavanje uvjeta zbog utvrđivanja uspješnog ili neuspješnog slanja/obrade poruke
- Omogućuje provođenje određenih akcija u ovisnosti o uspješnoj ili neuspješnoj obradi poruke, npr. slanje potvrdne obavijesti u slučaju uspjeha ili slanjem kompenzacijskih poruka u slučaju neuspjeha
- Komunikacija porukama uz uvjet pomiče odgovornost o uvjetima isporuke i obrade poruka iz primjenskog sustava u posrednički sustav