INTREBARI SPRC

1. **Prezentati cum se poate realiza propagarea actualizarilor pentru un depozit de date in doua tipuri diferite de sisteme: a) un sistem bazat pe un server central si b) un sistem fara server central, complet distribuit.**

   a) Intr-un sistem bazat pe un server central putem folosi protocoale push-based sau pull-based. Protocoalele push-based pot fi folosite pentru a asigura ca modificarile sunt transmise de server la toate replicile. Serverul central trimite mesaje de actualizare la fiecare copie. Sunt folosite atunci cand avem multe citiri. Protocoalele pull-based pot fi folosite pentru a preveni incarcarea prea mare a serverului. Replicile trimit cereri periodice la server pentru a verifica daca au aparut modificari. Aceasta strategie este folosita atunci cand sunt putine citiri.

   b) Propagarea actualizarilor in acest caz poate fi facuta folosind protocoalele epidemice. Protocoalele epidemice se bazeaza pe teoria epidemiilor si se utilizeaza pentru a propaga actualizarile la replici, folosind un numar minim de mesaje. Se transmit mesaje de la o replica la alta, pana cand toate sunt actualizate. Varianta gosspinig incearca sa evite nodurile deja actualizate. Interesul serverului scade pe masura ce intalneste servere actualizate deja. Varianta anti-entropie: serverele aleg la intamplare cui sa transmita si pot fi push, pull sau push-pull.
   Comparativ, protocoalele push-based ofera cel mai bun raport eficienta / cost, deoarece sunt capabile sa propage actualizarile la toate replicile intr-un timp scurt. In schimb, protocoalele epidemice sunt mai eficiente in cazul in care numarul de replici este mare si vrem sa micsoram numarul de mesaje transmise.

2. **Cum putem folsosi un commit distribuit pentru verificarea consistentei unui depozit de date cu n replici? Descrieti doua cazuri de utilizare pentru protocolul Three-Phase Commit.**

   Un commit distribuit se asigura ca toate replicile primesc si aplica aceleasi actualizari, in aceeasi ordine. Acest lucru se poate realiza prin utilizarea unui coordonator central. El este cel care initiaza operatia de commit si propaga cererea catre restul nodurilor. Intr-o astfel de retea se presupune ca ori fiecare proces executa operatia, ori niciunul. Acest model are 3 variante posibile, cele mai folosite fiind 2-phase commit si 3-phase commit. Coordonatorul trimite un mesaj „commit” tuturor replicilor, ceea ce le determină să aplice actualizările. Dacă oricare dintre replici raportează o inconsecvență, coordonatorul întrerupe tranzacția.

   Unul dintre cazuril de utilizare a lui Three-Phase Commit este managementul bazelor de date distribuite pentru a asigura că toate replicile din baza de date rămân consistente.
   Three-Phase Commit poate fi utilizat și în sistemele de procesare a tranzacțiilor distribuite, pentru a fii siguri că toate sistemele sunt de acord cu actualizările înainte de a efectua tranzacția. Acest lucru ajută la prevenirea inconsecvențelor și la asigurarea faptului că tranzacția este procesată atomic (fie toate actualizările sunt efectuate, fie niciuna dintre ele).

3. **Protocoale de consistenta. a) Ce sunt care este legatura lor cu modelele de consistenta? Protocoale bazate pe o copie primara: descriere succinta. Protocoale cu scriere replicata: descriere succinta; detalii despre cele bazate pe cvorum.**

   Un protocol de consistenta descrie o implementare a unui model de consistenta. Exista mai multe tipuri de astfel de protocoale.

   Unul dintre acestea este protocolul baza pe o copie primara, care este la randul sau impartit in protocol cu scriere la distanta(Remote Write), respectiv scriere locala (Local Write). Primul model implementeaza o consistenta secventiala, iar copia primara poate ordona operatiile de scriere primite. Cel de-al doilea presupune migrarea copiei primare la procesul care doreste sa faca actualizarea/scrierea.

   Protocolul cu scriere replicata are la baza modelul de replicare activa si presupune ca replicile au asociate un proces care executa operatiile de actualizare. De aceea, se foloseste un procedeu de multicast total ordonat, pentru a asigura ca operatiile sunt executate in aceeasi ordine in toate replicile. Prezinta probleme in cazul replicarii obiectelor.

   Ultimul model de protocol este cel bazat pe cvorumuri, 1 cvorum format din replici accesate la citire, respectiv la scriere. Are la baza 2 reguli pentru a asigura consistenta replicilor. Prima regula presupun insumarea cardinalelor celor 2 multimi trebuie sa fie strict mai mare decat numarul total de replici, pentru a ne asigura ca cel putin o replica de citire se afla si in cvorumul de scriere pentru a avea ultima versiune actualizata a datelor. A doua regula combate conflictele de tip write-write. Cardinalul multimii replicilor de scriere trebuie sa fie strict mai mare decat N/2(N : numarul total de replici). Prin urmare, cel putin o replica de scriere se afla in ambele cvorumuri de scriere. Daca cele 2 multimi ar fi disjuncte, asta ar conduce la replici cu acelasi numar de versiune, dar valori diferite ale datelor.

4. **Descrieti pe scurt modelul Harrison-Ruzzo-Ullman (HRU). Ce rezolva HRU si ce lasa nerezolvat? Cum putem sa folosim HRU intr-un sistem Peer-to-Peer?**

   Modelul Harrison-Ruzzo-Ullman este o varianta exstinsa a modelului Graham-Denning. Foloseste un set de subiecti S, un set de obiecte O si un set de drepturi R pentru a forma matricea de acces M, populata cu intrari Mso, ce reprezinta drepturile subiectului S asupra obiectului O. Daca toate conditiile dintr-o comanda sunt indeplinite, atunci se executa lista de operatii care va schimba starea matricei M in starea M'. HRU rezolva problema drepturilor de acces, deoarece asigura faptul ca subiectii pot avea acces doar la obiectele pentru care au drepturi.

   Cu toate acestea, este sensibil la scurgeri de drepturi, adica dintr-o stare a matricei M lasa sa se scurga dreptul r, daca exista o comanda c care adauga dreptul r intr-o intrare din M care anterior nu continea r. Prin urmare, proprietarul poate adauga drepturi de read/write, ceea ce nu este de dorit.

   La P2P, un nod poate garanta drepturi altui nod. De exemplu, o comanda numita "grant_access" poate fi definita pentru a permite unui nod sa acorde unui alt nod dreptul de a accesa informatia sa. O comanda "revoke_access" poate fi, de asemenea, definita pentru a revoca dreptul de acces de la un nod la altul.

5. **Descrieti principiul folosit in protocolul pentru managementul cheilor de grup - Diffie-Hellman key exchange. Discutati eficienta utilizarii protocolului pentru un grup peer-to-peer (general) cu n membri.**

   Modelul Diffie-Hellman key exchange pentru managementul cheilor de grup a fost introdus de Steiner si presupune existenta a n membri in grup, fiecare dintre acestia avand atribuit un secret xi. Se stabilesc 2 valori prime q si a si se calculeaza distribuit valori intermediare, de-a lungul a n runde. Primul membru va calcula a^x1 si va pasa rezultatul la cel de-al doilea, care la randul sau va calcula puterile a^x2 , respectiv a^x1*x2, si va genera cheia {a^x1,a^x2,a^x1.x2}. Procesul se repeta pana cand toti membrii isi vor calcula propriul set de valori intermediare, folosindu-se de ce a primit de la participantii anteteriori. Odata ajuns la final, la cel de-al n-lea membru, acesta calculeaza cheia k = a^x1.x2.x3…xn mod q, ridica restul valorilor intermediare la puterea secretului sau, xn, si transmite multicast tot setul, exceptand cheia k. Ulterior, fiecare membru i extrage din setul de valori pe aceea la care nu s-a folosit xi in ridicarea la putere si calculeaza propria sa cheie k = a^x1.x2.x3…xn mod q.

6. **Dati doua exemple de sisteme distribuite: unul in care componentele individuale au o vulnerabilitate mai mare decat a sistemului intreg si unul in care vulnerabilitatea sistemului este mai mare decat a componentelor individuale.**

   Un exemplu din prima categorie de sisteme distribuite este o infrastructură de cloud, in care mai multe servere sunt conectate și lucrează împreună pentru a oferi resurse de calcul utilizatorilor prin internet. Serverele individuale din infrastructura cloud pot fi vulnerabile la hacking sau la alte amenințări de securitate. Cu toate acestea, infrastructura cloud în ansamblu este proiectată pentru a fi extrem de redundantă și scalabilă, cu mai multe straturi de securitate și mecanisme de failover pentru a se asigura că sistemul continuă să funcționeze chiar dacă unul sau mai multe servere sunt compromise.

   Din a doua categorie face parte orice sistem centralizat cu un server principal, numit single point of failure, care, daca este compromis, intregul sistem devine vulnerabil. Spre exemplu, o banca cu mai multe ATM-uri, dar care are datele clientilor strocate intr-un singur loc.

7. **Explicati de ce modelul de protocoale bazate pe cvorum are nevoie de doua conditii de baza? Cu notatia folosita la curs, explicati ce semnificatie are conditia suplimentara N_R > N/2. Justificati daca aceasta este in contradictie cu celelalte 2 conditii.**

   Prima regula Nw + Nr > N presupune existenta a minim unei replici de citire in ambele cvorumuri pentru a asigura faptul ca macar o replica este actualizata in cvorumul de citire.

   Cea de-a doua regula Nw > N / 2 asigura faptul ca, in cazul in care 2 procese aleg in mod diferit cvorumurile proprii de replici de citire, cele 2 cvorumuri nu vor fi disjuncte si, prin urmare, chiar daca vor avea acelasi numar de versiune in urma a 2 scrieri succesive, vor avea garantat si aceleasi valori => consistenta.

Regula suplimentare Nr > N / 2 nu modifica cu nimic starea sistemului, in sensul ca nu este necesara pentru a rezolva vreo problema si nici nu cauzeaza vreuna noua.

8. **Explicati daca este posibil ca o memorie sa fie in acelasi timp si secvential si cauzal si FIFO consistenta. Dati un exemplu de utilizare pentru fiecare caz in parte (secvential, cauzal, FIFO).**
9. **Explicati de ce modelul de protocoale bazate pe cvorum are nevoie de doua conditii de baza? Cu notatia folosita la curs, explicati ce implicatie are inlocuirea conditiei N_W > N/2 cu N_W >= N/2.**

   Prima regula Nw + Nr > N presupune existenta a minim unei replici de citire in ambele cvorumuri pentru a asigura faptul ca macar o replica este actualizata in cvorumul de citire.

   Cea de-a doua regula Nw > N / 2 asigura faptul ca, in cazul in care 2 procese aleg in mod diferit cvorumurile proprii de replici de citire, cele 2 cvorumuri nu vor fi disjuncte si, prin urmare, chiar daca vor avea acelasi numar de versiune, vor avea garantat si aceleasi valori => consistenta.

   Daca cea de-a doua regula s-ar modifica la Nw >= N / 2, atunci, in cazul unui numar de procese mai mare sau egal cu 2, acestea ar putea alege multimi disjuncte de cvorumuri de scriere, daca s-ar alege Nw = N / 2. Prin urmare, s-ar ajunge din nou la cauza pentru care s-a adaugat cea dea doua regula, ceea ce nu este de dorit din cauza motivelor enumerate mai sus.

10. **Descrieti conceptele de scalabilitate si elasticitate in mediile Cloud. Descrieti doua situatii in care una din proprietati sa fie preferate celeilalte.**

    Scalabilitatea in mediul Cloud consta in faptul ca furnizorul poate oferi servicii de volum nelimitat. Oricati clienti ar trebui sa serveasca, se presupune ca un domeniu de Cloud are resursele necesare pentru a statisace tuturor cererile.

    Elasticitatea se refera la faptul ca resursele se pot aloca in functie de nevoile clientului, nu intr-un numar mai mare sau mai mic. Resursele pot fi adaugate la cresterea solicitarii din partea clientilor si sunt eliberate cand solicitarea scade.

    Scalabilitatea este preferata in cazul unei aplicatii care este activa un timp indelungat si este nevoie de un numar de resurse specific pentru a rula la parametrii maximi. Elasticitatea este utila in cazul unei aplicatii foarte solicitate intr-o anumita perioada a anului, precum un site de rezervari online de hoteluri/apartamente. O astfel de platforma va avea nevoie de mai multe resurse in timpul sezonului de vacante(vara), in timp ce in restul anului cererea tinde sa fie mai mica.

11. **Putem proiecta un sistem de posta electronica folosind RBAC?**

    RBAC(Role Based Access Control) este un model care asigura controlul accesului la resurse in sistemele distribuite. Utilizatorii au anumite roluri, iar fiecarui rol ii este asignat un set de permisiuni. In cazul unui sistem de posta electronica, fiecare ultilizator logat pe site va avea asociat rolul de client autorizat si autentificat si va avea un set de permisiuni asociate cu acesta, precum citirea mesajelor noi din cutia postala, stergerea acestora sau trimiterea de mesaje noi catre alti destinatari.