# Înțelegerea bazelor de date distribuite: arhitectură, provocări și aplicații

Gabriel Luican, Mihail Brazdis
Universitatea Transilvania din Brașov, Facultatea de Matematica si Informatica

Baze de Date Distribuite

Univ. Asist. Drd. Dragoș Tohănean

2024 Iunie

Autori: Gabriel Luican, Mihail Brazdiș
Contact: gabriel.luican@gmail.com, mbrazdis@gmail.com

**I. Introducere** **5**
A. Contextul bazelor de date 5
Definiția unei baze de date 5
Evoluția bazelor de date 5
Importanța bazelor de date distribuite 5
B. Scopul eseului 7
Domenii cheie care trebuie acoperite 7
C. Teza eseului 7
**II. Fundamentele bazelor de date distribuite** **8**
A. Definiție și concepte cheie 8
Noduri 8
Clustere 8
Shard-uri 9
Replici 9
B. Comparație cu bazele de date centralizate 9
Arhitectura 9
Scalabilitate 10
Fault Tolerance 10
Concluzie 10
C. Componentele bazelor de date distribuite 11
Noduri 11
Clustere 11
Partiționarea datelor 11
Replicare 11
Algoritmi de consens 11
**III. Managementul Tranzacțiilor** **13**
A. Proprietăţile ACID în sistemele distribuite 13
Atomicity 13
Consistency 13
Isolation 14
Durability 14
B. Protocolul de angajare în două faze 14
C. Tranzacții distribuite 15
**IV. Toleranță la erori și disponibilitate ridicată** **17**
A. Detectarea și recuperarea erorilor 17
Redundancy 17
Failover 17

Load Balancing 17
B. Redundanța datelor 17
Funcționalități și caracteristici 18
Beneficii și cazuri de utilizare 18
Provocări și limitări 18
Integrare cu Data Lakehouse 18
**V. Securitatea în bazele de date distribuite 20**
A. Securitatea datelor 20
Autentificare și autorizare 20
Criptarea datelor 20
Intrare validată 20
B. Controlul accesului 20
Securizarea datelor în repaus cu criptare 20
Importanța criptării datelor în repaus 21
Controlul accesului și autentificarea 21
C. Audit și Monitorizare 21
Advanced Encryption Standard (AES) 21
**VI. Referințe 22**

# I. Introducere

## A. Contextul bazelor de date

### *Definiția unei baze de date*

În tehnologia informației, o **bază de date** este o colecție organizată de date sau un tip de depozit de date bazat pe utilizarea unui **sistem de management al bazei de date** (**SGBD**), software-ul care interacționează cu utilizatorii finali, aplicațiile și baza de date în sine pentru a captura și analiza datele. SGBD-ul cuprinde în plus facilitățile de bază furnizate pentru administrarea bazei de date. Suma totală a bazei de date, a SGBD-ului și a aplicațiilor asociate poate fi denumită un sistem de baze de date. Adesea, termenul „bază de date" este folosit și pentru a se referi la oricare dintre SGBD, sistemul de baze de date sau o aplicație asociată cu baza de date.

### *Evoluția bazelor de date*

Bazele de date au evoluat dramatic de la înființarea lor la începutul anilor 1960. **Bazele de date de navigație**, cum ar fi baza de date ierarhică (care se baza pe un model arborescent și permitea doar o relație unu la mai mulți) și baza de date în rețea (un model mai flexibil care permitea relații multiple), au fost sistemele originale folosite pentru a stoca și să manipuleze datele. Deși simple, aceste sisteme timpurii erau inflexibile. În anii 1980, **bazele de date relaționale** au devenit populare, urmate de **bazele de date orientate pe obiecte** în anii 1990. Mai recent, **bazele de date NoSQL** au apărut ca răspuns la creșterea internetului și nevoia de viteză mai rapidă și procesare a datelor nestructurate. Astăzi, bazele de date în cloud și bazele de date autonome lansează noi drumuri când vine vorba de modul în care datele sunt colectate, stocate, gestionate și utilizate.

### *Importanța bazelor de date distribuite*

Bazele de date distribuite oferă mai multe avantaje față de bazele de date centralizate tradiționale, făcându-le o alegere populară pentru aplicațiile moderne.

**Scalabilitate:** Unul dintre avantajele cheie ale bazelor de date distribuite este capacitatea lor de a se scala pe orizontală. Pe măsură ce cantitatea de date din baza de date crește, pot fi

adăugate noi noduri la cluster pentru a gestiona sarcina crescută. Acest lucru permite aplicațiilor să se scaleze rapid și ușor, fără a fi nevoie de upgrade-uri hardware costisitoare sau de timpi de nefuncționare.

**Disponibilitate ridicată:** bazele de date distribuite oferă o disponibilitate ridicată prin replicarea datelor pe mai multe noduri. Dacă un nod eșuează, baza de date poate continua să funcționeze prin direcționarea cererilor către alte noduri din cluster. Acest lucru asigură că aplicația poate continua să funcționeze chiar și în cazul unei defecțiuni hardware sau a unei întreruperi de rețea.

**Toleranță la erori:** bazele de date distribuite sunt tolerante la erori, ceea ce înseamnă că pot continua să funcționeze chiar dacă mai multe noduri eșuează. Acest lucru se realizează prin replicarea datelor în mai multe noduri și utilizarea algoritmilor de consens pentru a se asigura că datele rămân consecvente în cluster.

**Localitatea datelor:** bazele de date distribuite pot fi proiectate pentru a stoca date în locații mai apropiate de utilizatori sau aplicații care au nevoie de ele. Acest lucru poate ajuta la reducerea latenței și la îmbunătățirea performanței aplicației, reducând la minimum timpul necesar pentru a prelua datele din baza de date.

**Cost-eficiente:** bazele de date distribuite pot fi mai rentabile decât bazele de date centralizate, mai ales pe măsură ce cantitatea de date din baza de date crește. Prin utilizarea hardware-ului de bază și a software-ului open source, bazele de date distribuite pot fi construite și operate la un cost mai mic decât bazele de date centralizate tradiționale.

**Flexibilitate:** bazele de date distribuite pot fi personalizate pentru a satisface nevoile specifice ale unei aplicații. Diferite modele de date și motoare de stocare pot fi utilizate pentru a optimiza performanța și scalabilitatea pentru cerințele unice ale aplicației.

**Disponibilitate globală:** bazele de date distribuite pot fi proiectate pentru a oferi disponibilitate globală prin replicarea datelor în mai multe centre de date sau puncte de prezență

situate în diferite regiuni ale lumii. Acest lucru poate ajuta la îmbunătățirea performanței aplicației pentru utilizatorii aflați în diferite părți ale lumii, reducând la minimum latența asociată cu preluarea datelor dintr-o locație centralizată.

**B. Scopul eseului**

Scopul acestui eseu este de a aprofunda în importanța și relevanța bazelor de date distribuite în calculul modern. Acesta își propune să ofere o înțelegere cuprinzătoare a bazelor de date, a evoluției lor și a modului în care acestea s-au adaptat pentru a răspunde cerințelor erei digitale. Eseul va explora conceptul de baze de date distribuite, avantajele acestora și de ce au devenit o piatră de temelie în domeniul managementului datelor.

***Domenii cheie care trebuie acoperite***

Eseul va începe prin definirea bazelor de date și a rolului lor fundamental în stocarea și gestionarea datelor. Apoi va urmări evoluția bazelor de date de la formele lor timpurii până la structurile mai complexe pe care le vedem astăzi. Această perspectivă istorică va oferi un context pentru înțelegerea apariției și semnificației bazelor de date distribuite.

Corpul principal al eseului se va concentra pe bazele de date distribuite, explicând structura, funcționalitatea și beneficiile acestora. Se va discuta cum bazele de date distribuite abordează limitările bazelor de date tradiționale, în special în ceea ce privește scalabilitatea, disponibilitatea și toleranța la erori. Eseul va evidenția, de asemenea, rolul bazelor de date distribuite în sprijinirea aplicațiilor moderne, în special a celor care operează în cloud.

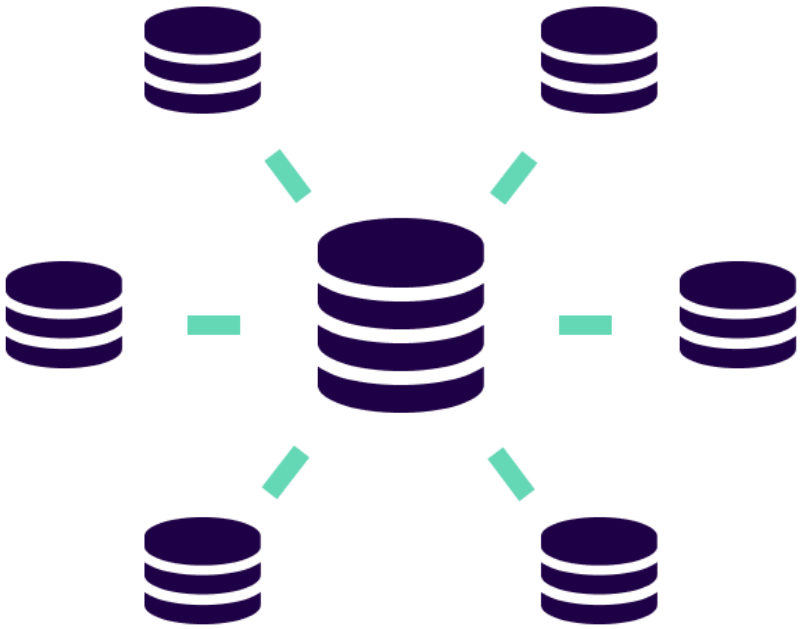
**C. Teza eseului**

Prin acest eseu, cititorul va dobândi o înțelegere solidă a bazelor de date distribuite și a rolului lor critic în calculul modern. Eseul nu numai că va informa, ci și va stimula gândirea despre viitorul managementului datelor într-o lume din ce în ce mai digitală.

## II. Fundamentele bazelor de date distribuite

### A. Definiție și concepte cheie

O **bază de date distribuită** este o bază de date în care datele sunt stocate în diferite locații fizice. Poate fi stocat în mai multe computere situate în aceeași locație fizică (de exemplu, un **centru de date**); sau poate dispersate într-o rețea de calculatoare interconectate. Spre deosebire de sistemele paralele, în care procesoarele sunt strâns cuplate și constituie un singur sistem de bază de date, un sistem de baze de date distribuite constă din site-uri cuplate slab care nu au componente fizice în comun.



***Noduri***

Nodurile sunt instanțe individuale ale Elasticsearch care rulează pe o mașină care participă la un cluster. Fiecare nod stochează date și participă la capacitățile de indexare și căutare ale clusterului. Există diferite tipuri de noduri, inclusiv:

- **Nodul principal:** controlează clusterul, gestionează modificările la nivel de cluster și gestionează acțiuni precum crearea sau ștergerea unui index.
- **Nod de date:** stochează date și execută operațiuni legate de date, cum ar fi căutarea și indexarea.
- **Nod client:** direcționează cererile de la clienți către nodurile corespunzătoare din cluster.

**Caz de utilizare:** Nodurile sunt similare membrilor individuali ai echipei dintr-un cluster. Fiecare membru (nod) contribuie la efortul colectiv prin stocarea și procesarea datelor în timp ce colaborează cu alți membri (noduri) pentru o funcționare eficientă.

***Clustere***

Un cluster din este o colecție de unul sau mai multe noduri (servere) care dețin datele tale întregi și oferă capabilități de indexare și căutare în toate nodurile.

Clusterele sunt utilizate pentru a obține o disponibilitate ridicată și toleranță la erori. Ei distribuie datele pe mai multe noduri, asigurându-se că, dacă un nod eșuează, datele rămân accesibile de la alte noduri.

**Caz de utilizare**: Imaginați-vă un cluster ca o echipă de computere interconectate care colaborează pentru a stoca și procesa date. Fiecare nod contribuie la funcționarea generală a clusterului și, în mod colectiv, se ocupă de stocarea datelor și operațiunile de căutare.

***Shard-uri***

Shard-urile sunt subseturi mai mici ale unui index care stochează o parte din datele indexului. Elasticsearch împarte indecșii în mai multe shard-uri pentru a distribui datele și a permite procesarea paralelă. Există diferite tipuri de cioburi, inclusiv:

- Shard-uri primare: copia originală a datelor care poate fi replicată în continuare pentru toleranță la erori.
- Shard-uri replica: copii suplimentare ale shard-urilor primare utilizate pentru redundanță și performanță îmbunătățită de căutare.

**Caz de utilizare:** Gândiți-vă la shard-uri ca la segmente ale unui proiect mare. Prin împărțirea unui proiect în segmente mai mici, fiecare membru al echipei (nod) poate lucra pe diferite segmente simultan, permițând finalizare mai rapidă și o toleranță mai bună la erori.

### *Replici*

Replicile sunt copii ale shard-urilor primare, care servesc ca mecanisme de failover. Acestea îmbunătățesc rezistența sistemului și permit operațiuni paralele de căutare și recuperare.

## B. Comparație cu bazele de date centralizate

Bazele de date centralizate și distribuite diferă semnificativ în ceea ce privește arhitectura, scalabilitatea și toleranța la erori.

### *Arhitectura*

| **Baze de date centralizate** | **Baze de date distribuite** |
|---|---|
| O bază de date centralizată este stocată, localizată și întreținută într-o singură locație. Este accesat, modificat și gestionat chiar din acea locație. Acest tip de bază de date este utilizat de obicei de instituții sau organizații. | O bază de date distribuită constă din mai multe baze de date care sunt conectate între ele și sunt răspândite în diferite locații fizice. Datele stocate în diferite locații fizice pot fi gestionate independent de alte locații fizice. |

### *Scalabilitate*

| **Baze de date centralizate** | **Baze de date distribuite** |
|---|---|
| Timpul de acces la date în cazul mai multor utilizatori este mai mult într-o bază de date centralizată. Este mai ieftin în comparație cu toate celelalte baze de date disponibile. | Această bază de date poate fi extinsă cu ușurință, deoarece datele sunt deja răspândite în diferite locații fizice. Timpul de acces la date în cazul mai multor utilizatori este mai mic într-o bază de date distribuită. Cu toate acestea, această bază de date este foarte costisitoare și este dificil de întreținut din cauza complexității sale. |

***Fault Tolerance***

| **Baze de date centralizate** | **Baze de date distribuite** |
|---|---|
| Dacă apare orice fel de defecțiune a sistemului în sistemul centralizat, atunci toate datele vor fi distruse. | Această bază de date este mai sigură în comparație cu o bază de date centralizată. Dacă o bază de date eșuează, utilizatorii pot accesa sistemul prin alte fișiere. |

***Concluzie***

Pe scurt, bazele de date distribuite oferă scalabilitate, toleranță la erori și performanță îmbunătățită, dar necesită mecanisme de gestionare și coordonare mai complexe și pot implica costuri mai mari de instalare și întreținere. Bazele de date centralizate și distribuite au fiecare propriile atribute unice și sunt potrivite pentru diferite cazuri de utilizare. Bazele de date distribuite au fiecare propriile atribute unice și sunt potrivite pentru diferite cazuri de utilizare.

**C. Componentele bazelor de date distribuite**

***Noduri***

Nodurile sunt servere sau computere individuale care locuiesc într-un sistem de baze de date distribuite. Fiecare nod stochează un set de date și rulează pe software-ul sistemului de management al bazelor de date distribuite (DDBMS).

***Clustere***

Un cluster dintr-o bază de date distribuită este o colecție de unul sau mai multe noduri care dețin toate datele tale și oferă capabilități de indexare și căutare în toate nodurile. Clusterele sunt utilizate pentru a obține o disponibilitate ridicată și toleranță la erori. Ei distribuie datele pe mai multe noduri, asigurându-se că, dacă un nod eșuează, datele rămân accesibile de la alte noduri.

***Partiționarea datelor***

Este procesul de împărțire a unui set mare de date în subseturi mai mici, mai ușor de gestionat, numite partiții. Criteriile de partiţionare a datelor şi strategia de partiţionare decid

modul în care este împărţit setul de date. Fiecare partiție conține un subset de date, iar aceste subseturi sunt de obicei distribuite pe mai multe servere, noduri sau sisteme. Principalele obiective ale partiționării datelor sunt îmbunătățirea performanței, scalabilității și toleranței la erori.

### *Replicare*

Replicarea bazei de date este procesul de copiere a datelor și de stocare a acestora în diferite locații. Efectuarea replicării datelor asigură că există o copie consecventă a bazei de date în toate nodurile dintr-un sistem distribuit. Acest lucru servește pentru a face datele disponibile pe scară largă și pentru a proteja împotriva pierderii de date.

### *Algoritmi de consens*

Se referă la protocoalele pe care sistemele distribuite le folosesc pentru a ajunge la un acord asupra unei singure valori de date sau a unei ordini de operațiuni între procesele sau nodurile distribuite, în ciuda potențialelor defecțiuni ale procesului, partițiilor de rețea sau defecțiunilor bizantine.

Aceste componente sunt fundamentale pentru înțelegerea modului în care funcționează bazele de date distribuite și a modului în care oferă beneficii precum scalabilitate, toleranță la erori și disponibilitate ridicată.

## III. Managementul Tranzacțiilor

### A. Proprietăţile ACID în sistemele distribuite

*Asigurarea atomicității, consistenței, izolației și durabilității.*

Acronimul ACID înseamnă Atomicity, Consistency, Isolation, and Durability. Aceste patru proprietăți ACID definesc modul în care tranzacțiile de bază de date ar trebui să se comporte pentru a se asigura că datele rămân într-o stare consecventă chiar și în cazul unei defecțiuni a sistemului.

***Atomicity***

Tratează o întreagă tranzacție ca pe o singură unitate de lucru. Tranzacțiile cu baze de date implică, de obicei, mai multe operațiuni de bază de date. Dacă oricare dintre ele eșuează, parțial sau complet, atunci rezultatul final al tranzacției poate fi incorect. Tranzacțiile atomice necesită finalizarea cu succes a fiecărui pas operațional. Dacă o parte a tranzacției eșuează, întreaga tranzacție eșuează și trebuie să ruleze din nou.

De exemplu, transferul de fonduri dintr-un cont bancar în altul presupune scăderea sumei de transfer din primul și adăugarea acesteia la al doilea. Dacă oricare dintre etape eșuează, atunci

cel puțin un sold de cont va fi greșit. O tranzacție atomică care se confruntă cu un eșec operațional similar nu ar avea loc deloc.

***Consistency***

Este aplicarea regulilor de afaceri și a constrângerilor de integritate a datelor care guvernează modul în care tranzacțiile schimbă starea unei baze de date. Baza de date ar deveni nesigură dacă tranzacțiile ar fi permise să încalce aceste restricții. Nu ar exista nicio modalitate de a spune dacă oricare două valori ale datelor sunt comparabile.

De exemplu, o sumă de retragere care depășește soldul unui cont ar încălca regula unei bănci care interzice descoperirile de cont. Consecvența împiedică finalizarea tranzacției, revenind sistemul la starea anterioară.

***Isolation***

Tranzacțiile concurente, cele care interacționează cu aceleași date în același timp, ar putea submina integritatea datelor. Standardul ANSI/ISO SQL descrie trei situații care apar în funcție de momentul operațiunilor a două tranzacții:

- **Citiri Dirty**: Tranzacția A a actualizat un rând, dar nu a comis încă modificarea atunci când tranzacția B preia rândul.
- **Citiri Non-repeatable:** Tranzacția B preia un rând, tranzacția A comite actualizări pentru rând, iar tranzacția B preia din nou rândul.
- **Citiri Phantom**: Tranzacția B preia un set de rânduri, tranzacția A inserează sau elimină rânduri din acel set, iar tranzacția B preia din nou setul de rânduri.

Sistemele compatibile cu ACID folosesc controale de concurență pentru a izola tranzacțiile unele de altele. Controalele bazate pe blocare forțează o nouă tranzacție să aștepte finalizarea tranzacției curente. Controalele cu versiuni multiple folosesc izolarea instantanee, permițând noii tranzacții să acționeze asupra stării curente în timp ce cealaltă tranzacție funcționează. Controlul angajează modificările noii tranzacții cu condiția să nu existe conflict; în caz contrar, respinge tranzacția.

Sistemele de baze de date oferă diferite niveluri de izolare care echilibrează izolarea și performanța. Aceste niveluri dictează modul în care au loc operațiunile de citire și scriere și dacă se permit citirile phantom, non-repeatable sau dirty.

***Durability***

Înseamnă pur și simplu că, odată ce o tranzacție își comite modificările, acele modificări devin parte din înregistrarea permanentă a bazei de date, chiar și în cazul unei întreruperi de curent sau a altor defecțiuni ale sistemului. Sistemele de baze de date realizează de obicei durabilitate prin mutarea datelor din memorie în stocarea nevolatilă.

## B. Protocolul de angajare în două faze

Explicații și provocări. Protocol de comitere în două faze (gestionarea tranzacțiilor distribuite)

Luați în considerare că ni se oferă un set de magazine alimentare în care șeful tuturor magazinelor dorește să întrebe despre inventarul de dezinfectanți disponibil în toate magazinele pentru a muta stocul magazin în magazin pentru a echilibra cantitatea de dezinfectanți din toate magazinele. Sarcina este efectuată de o singură tranzacție T care este componenta Tn la nthstore și un magazin S0 corespunde lui T0 unde se află managerul. Următoarea secvență de activități sunt efectuate de T sunt mai jos:

1. Componenta tranzacției(T) $T_0$ este creată la sediul central (sediul central).
2. $T_0$ trimite mesaje către toate magazinele pentru a le comanda să creeze componente $T_i$.
3. Fiecare $T_i$ execută o interogare la magazinul „i” pentru a descoperi cantitatea de dezinfectanți disponibil și raportează acest număr către $T_0$.

## C. Tranzacții distribuite

*Cum funcționează tranzacțiile distribuite?*

Tranzacțiile distribuite au aceleași cerințe de finalizare a procesării ca și tranzacțiile obișnuite de baze de date, dar trebuie gestionate prin mai multe resurse, ceea ce le face mai dificil de implementat pentru dezvoltatorii de baze de date. Resursele multiple adaugă mai multe puncte de defecțiune, cum ar fi sistemele software separate care rulează resursele (de exemplu, software-ul bazei de date), serverele hardware suplimentare și defecțiunile rețelei. Acest lucru

face ca tranzacțiile distribuite să fie susceptibile la eșecuri, motiv pentru care trebuie puse măsuri de protecție pentru a păstra integritatea datelor.

Pentru ca o tranzacție distribuită să aibă loc, managerii de tranzacții coordonează resursele (fie mai multe baze de date, fie mai multe noduri ale unei singure baze de date). Managerul de tranzacții poate fi unul dintre depozitele de date care vor fi actualizate ca parte a tranzacției sau poate fi o resursă separată complet independentă, care este responsabilă doar de coordonare. Managerul de tranzacții decide dacă să comite o tranzacție de succes sau să anuleze o tranzacție nereușită, cea din urmă lăsând baza de date neschimbată.

În primul rând, o aplicație solicită tranzacția distribuită managerului de tranzacții. Managerul de tranzacții se ramifică apoi la fiecare resursă, care va avea propriul „manager de resurse” pentru a o ajuta să participe la tranzacțiile distribuite. Tranzacțiile distribuite sunt adesea efectuate în două faze pentru a proteja împotriva actualizărilor parțiale care ar putea apărea atunci când se întâlnește o defecțiune. Prima fază implică recunoașterea unei intenții de a se angaja sau a unei faze de „pregătire pentru angajare”. După ce toate resursele sunt recunoscute, li se cere apoi să execute un commit final, iar apoi tranzacția este finalizată.

## IV. Toleranță la erori și disponibilitate ridicată

### A. Detectarea și recuperarea erorilor

*Tehnici pentru construirea de sisteme tolerante la erori: asigurarea operațiunilor continue și a rezistenței.*

Există mai multe elemente și tehnici cheie implicate în atingerea toleranței la erori:

***Redundancy***

Este un aspect fundamental al toleranței la erori. Aceasta implică duplicarea componentelor sau sistemelor critice pentru a crea copii de rezervă care pot prelua fără probleme în cazul unei defecțiuni. Având componente redundante, sistemul poate continua să funcționeze fără întreruperi. De exemplu, într-un cluster de servere tolerant la erori, mai multe servere sunt configurate pentru a gestiona sarcina de lucru, iar dacă un server eșuează, celelalte pot interveni imediat pentru a asigura un serviciu neîntrerupt.

***Failover***

Este procesul de comutare automată la un sistem sau o componentă redundantă atunci când este detectată o defecțiune. Se asigură că sistemul de rezervă preia fără probleme și continuă să ofere serviciile necesare. Mecanismele de failover sunt utilizate în mod obișnuit în infrastructura de rețea, unde routerele sau switch-urile pot trece la dispozitive redundante fără a afecta conectivitatea rețelei.

***Load Balancing***

Load balancing-ul distribuie sarcina de lucru pe mai multe sisteme sau componente pentru a preveni copleșirea oricărei componente. Prin distribuirea uniformă a sarcinii, toleranța la defecțiuni este îmbunătățită, deoarece reduce riscul ca componentele individuale să fie supraîncărcate sau să se defecteze din cauza stresului excesiv. Load balacers pot direcționa în mod inteligent cererile primite către resursele disponibile, optimizând performanța și minimizând impactul defecțiunilor.

### B. Redundanța datelor

*Ce este redundanța datelor?*

Redundanța datelor se referă la stocarea aceleiași date în mai multe locuri, fie într-o singură bază de date, fie în mai multe sisteme de date. Deși poate părea a fi o risipă de spațiu de stocare, are roluri vitale în ceea ce privește fiabilitatea datelor, toleranța la erori și performanța sistemului.

***Funcționalități și caracteristici***

Datele redundante pot servi ca rezervă în timpul defecțiunilor sistemului sau ștergerii accidentale a datelor. Îmbunătățește disponibilitatea datelor, deoarece aceleași date pot fi preluate din mai multe locații, asigurând operațiuni continue chiar și atunci când o sursă de date eșuează. Deși are ca rezultat creșterea nevoilor de stocare, soluțiile moderne de stocare au făcut ca acest cost să fie neglijabil în comparație cu beneficiile.

***Beneficii și cazuri de utilizare***

**Disponibilitate îmbunătățită a datelor:** Datele sunt accesibile din mai multe locații chiar dacă o parte a sistemului se defectează**.**

**Toleranță la erori:** În cazul pierderii accidentale a datelor, datele sunt păstrate în altă parte.

**Performanță crescută a sistemului:** Copiile multiple ale datelor pot gestiona mai multe solicitări simultane, rezultând timpi de răspuns mai rapid.

***Provocări și limitări***

Redundanța datelor poate duce la inconsecvențe ale datelor dacă nu este gestionată corespunzător. Când există mai multe copii ale datelor, actualizarea datelor în toate locurile simultan poate fi o provocare. De asemenea, datele redundante necesită spațiu de stocare suplimentar, ceea ce poate duce la creșterea costurilor operaționale.

***Integrare cu Data Lakehouse***

Un data Lakehouse combină caracteristicile depozitelor tradiționale de date și ale lacurilor de date moderne, oferind o platformă unificată pentru tot felul de operațiuni de date. Redundanța datelor într-un data Lakehouse poate ajuta la preluarea și procesarea rapidă a datelor.

Cu toate acestea, un lac de date bine structurat ar trebui să minimizeze redundanța datelor și să folosească strategii pentru a menține consistența datelor.

## V. Securitatea în bazele de date distribuite

### A. Securitatea datelor

În sistemele distribuite, este imperativ să se adopte măsuri pentru a securiza datele în afară de comunicații. Măsurile de securitate a datelor sunt:

***Autentificare și autorizare***

Acestea sunt măsurile de control al accesului adoptate pentru a se asigura că numai utilizatorii autentici pot utiliza baza de date. Pentru a oferi autentificarea se folosesc certificate digitale. În plus, autentificarea este restricționată prin combinația nume de utilizator/parolă.

***Criptarea datelor***

Cele două abordări pentru criptarea datelor în sistemele distribuite sunt:

**Abordare internă către baza de date distribuită:** aplicațiile utilizator criptează datele și apoi stochează datele criptate în baza de date. Pentru utilizarea datelor stocate, aplicațiile preiau datele criptate din baza de date și apoi le decriptează.

**Extern la baza de date distribuită:** sistemul de baze de date distribuite are propriile sale capabilități de criptare. Aplicațiile utilizator stochează date și le preiau fără a realiza că datele sunt stocate într-o formă criptată în baza de date.

***Intrare validată***

În această măsură de securitate, aplicația utilizator verifică fiecare intrare înainte de a putea fi utilizată pentru actualizarea bazei de date. O intrare nevalidată poate provoca o gamă largă de exploit-uri, cum ar fi buffer overrun, command injection, cross-site scripting și coruperea datelor.

### B. Controlul accesului

***Securizarea datelor în repaus cu criptare***

Datele în repaus se referă la informațiile care sunt stocate și salvate pe o unitate de stocare fizică, cum ar fi hard disk-uri, unități SSD și alte dispozitive de stocare. Aceste date nu sunt utilizate sau transmise în mod activ. Chiar dacă este posibil ca datele să nu fie în mișcare, sunt totuși vulnerabile la accesul neautorizat, mai ales dacă dispozitivul de stocare este pierdut,

furat sau compromis. Exemple de date în repaus includ fișierele stocate pe hard diskul unui computer, datele stocate pe o unitate USB sau informațiile salvate într-o bază de date.

### *Importanța criptării datelor în repaus*

Criptarea datelor în repaus este esențială pentru protejarea informațiilor sensibile împotriva accesului neautorizat. Fără criptare, dacă un utilizator rău intenționat obține acces fizic la dispozitivul de stocare, poate citi și fura cu ușurință date sensibile. Criptarea transformă datele într-un format imposibil de citit, care poate fi descifrat numai cu cheia de decriptare adecvată. Acest lucru adaugă un strat suplimentar de protecție și asigură că, chiar dacă dispozitivul de stocare este compromis, datele rămân în siguranță.

### *Controlul accesului și autentificarea*

Aplicați controale puternice de acces și mecanisme de autentificare. Doar utilizatorii autorizați cu acreditări de autentificare adecvate ar trebui să poată accesa datele criptate. Autentificarea cu mai mulți factori adaugă un nivel suplimentar de securitate.

## C. Audit și Monitorizare

*Instrumente și practici pentru asigurarea integrității și securității datelor.*

Există diverse tehnologii și instrumente de criptare care sunt utilizate în mod obișnuit pentru a securiza datele, comunicațiile și rețelele. Aceste metode de criptare joacă un rol crucial în asigurarea confidențialității și integrității informațiilor sensibile.

### *Advanced Encryption Standard (AES)*

Advanced Encryption Standard (AES) este o metodă adoptată pe scară largă pentru păstrarea în siguranță a datelor prin conversia lor într-o formă codificată care poate fi înțeleasă doar cu cheia de decriptare corectă. Gândiți-vă la el ca la un cod secret care blochează și deblochează informații. AES poate fi asemănat cu o încuietoare digitală care utilizează o cheie specifică pentru a securiza și decripta datele.

## VI. Referințe


1. Ullman, Jeffrey; Widom, Jennifer (1997). A First Course in Database Systems. Prentice–Hall. ISBN 978-0138613372, online https://en.wikipedia.org/wiki/Database
2. Bachman, Charles W. (1973). "The Programmer as Navigator". Communications of the ACM. 16 (11): 653–658. doi:10.1145/355611.362534, online https://en.wikipedia.org/wiki/Database
3. Chong, Raul F.; Wang, Xiaomei; Dang, Michael; Snow, Dwaine R. (2007). "Introduction to DB2". Understanding DB2: Learning Visually with Examples (2nd ed.). IBM Press Pearson plc. ISBN 978-0131580183. Retrieved 17 March 2013, online https://en.wikipedia.org/wiki/Database
4. Undisclosed Author, "What Is a Database?", Oracle.com (November 24, 2020) online https://www.oracle.com/database/what-is-database/
5. Undisclosed Author, "Advantages of Distributed Databases for Modern Applications", Macrometa.com (Undisclosed date), online https://www.macrometa.com/articles/advantages-of-distributed-databases-for-modern-applications
6. Irina Linnik, "Distributed Databases Explained: How Exactly Do They Work", Softteco.com (January 30, 2024), online https://softteco.com/blog/what-is-a-distributed-database
7. Jatin Sharma, "Data Encryption: Securing Data at Rest and in Transit with Encryption Technologies" (August 16, 2023), online https://dev.to/documatic/data-encryption-securing-data-at-rest-and-in-transit-with-encryption-technologies-1lc2#securing-data-at-rest-with-encryption
8. Undisclosed Author, "DDBMS - Security in Distributed Databases" (Undisclosed date), online https://www.tutorialspoint.com/distributed_dbms/distributed_dbms_security_distributed_databases.htm

9. Undisclosed Author, “What is Data Redundancy” (Undisclosed date), online https://www.dremio.com/wiki/data-redundancy/
10. Undisclosed Author, “Comparing High Availability vs Fault Tolerance vs Disaster Recovery” (Undisclosed date), online Comparing High Availability Vs Fault Tolerance Vs Disaster Recovery (stonefly.com)
11. Undisclosed Author, “What is a Distributed Transaction?” (Undisclosed date), online What is a Distributed Transaction? | Hazelcast
12. Madhav_mohan, “Two Phase Commit Protocol (Distributed Transaction Management)” (29 Dec, 2020), online Two Phase Commit Protocol (Distributed Transaction Management) - GeeksforGeeks
13. Undisclosed Author “ACID Transactions” (Undisclosed date), online ACID Transactions: Atomicity, Consistency, Isolation, Durability (starburst.io)