

Capitolul 9

Sisteme NoSQL

F. Radulescu. Curs: Baze de date 1

1

NoSQL?

- ◆ Denumirea de sisteme de tip SQL este data sistemelor traditionale de gestiune a bazelor de date, bazate pe modelul relational al datelor
- ◆ Acestea sunt foarte bune pentru sistemele operationale ale unei organizatii, companii, etc. dar nu sunt cea mai buna solutie pentru alte tipuri de aplicatii, cum ar fi gestionarea unui volum mare de date.

F. Radulescu. Curs: Baze de date 1

2

NoSQL?

- ◆ Initial, NoSQL desemna folosirea unor alte solutii de stocare a datelor, diferite de cele clasice, si care nu utilizau limbajul SQL.
- ◆ A aparut apoi o nuanțare a acestei acceptiuni: actualmente NoSQL desemneaza solutii de stocare care pot sa imbine stocarea clasica, relationala, cu alte metode de stocare.
- ◆ Acum NoSQL = Not only SQL

F. Radulescu. Curs: Baze de date 1

3

Diferente

- ◆ Exista o serie de diferente intre un sistem SQL si unul NoSQL.
- ◆ Se poate face o comparatie intre cele doua tipuri de sisteme prin prisma mai multor caracteristici care pot fi (vezi si [Widom, 1]):

F. Radulescu. Curs: Baze de date 1

4

Diferente

- ◆ Caracteristici:
 1. Adaptare la necesitatile aplicatiilor
 2. Multi-user
 3. Acces securizat
 4. Persistenta datelor
 5. Siguranta in functionare
 6. Lucrul cu masive de date
 7. Eficienta

F. Radulescu. Curs: Baze de date 1

5

1. Adaptare ...

- ◆ In cazul unui sistem clasic (SQL), acesta are:
 - ◆ Un model simplu de stocare (date sub forma de tabele)
 - ◆ Un limbaj de cereri declarativ (cum este SQL)
 - ◆ Mecanisme de garantare a tranzactiilor.
- ◆ Nu intotdeauna insa toate aceste caracteristici sunt utile sau usor de folosit.

F. Radulescu. Curs: Baze de date 1

6

Model simplu

- ◆ In ceea ce priveste simplitatea modelului: nu intotdeauna datele de stocat au o forma tabelara in mod natural.
- ◆ In acest caz este necesara preprocesarea lor pentru a le aduce la aceasta forma.

Limbaj declarativ

- ◆ Limbajul de cereri declarativ – SQL – este un plus in cazul aplicatiilor de tip întreprindere dar are si dezavantaje.
- ◆ Fiind un limbaj complet, care poate efectua intreaga gama de operatii, aduce cu sine un overhead de sistem care poate fi semnificativ

Limbaj declarativ

- ◆ Exista o serie de aplicatii care nu au nevoie decat de o mica parte a facilitatilor unui astfel de limbaj – de exemplu doar cautarea si incarcarea de perechi cheie valoare.
- ◆ De aceea sistemele NoSQL pun la dispozitie doar strictul necesar in aceasta privinta (eventual nici nu au un limbaj de cereri ci doar un API).

Garantarea tranzactiilor

- ◆ Uneori aceasta cerinta nu este atat de importanta – de exemplu in cazul stocarii datelor pentru un motor de cautare unde operatiile sunt foarte simple.
- ◆ Overhead-ul unui sistem SQL reprezinta in acest caz un dezavantaj.
- ◆ Sistemele NoSQL pot avea facilitati mai reduse in aceasta privinta.

2. Multi-user

- ◆ In cazul unor aplicatii – de exemplu stocarea datelor pentru motoarele de cautare, deja mentionata – nu este cazul de a lua atat de multe masuri pentru pastrarea consistentei acestora in cazul accesului simultan.
- ◆ Un sistem clasic (SQL) consuma in acest caz multe resurse inutil.

3. Acces securizat

- ◆ Aceasta caracteristica este utila atat in cazul sistemelor traditionale (SQL) cat si in cazul sistemelor NoSQL.
- ◆ Exista insa o serie de aplicatii in care nu este atat de importanta, iar sistemele NoSQL au mecanisme mai putin complicate pentru accesul utilizatorilor
- ◆ Un exemplu: in cazul in care prelucrarile se fac off-line.

4. Persistenta datelor

- ◆ Această caracteristică este importantă atât pentru sisteme SQL cât și pentru cele NoSQL.
- ◆ În cazul celor din urmă însă nu sunt întotdeauna necesare mecanisme complicate de organizare a datelor pe disc, sistemele NoSQL putând utiliza inclusiv doar fișiere simple.

F. Radulescu, Curs: Baze de date I

13

5. Siguranța în funcționare

- ◆ Și această caracteristică este importantă atât pentru sisteme SQL cât și pentru cele NoSQL.
- ◆ Există însă cazuri în care se lucrează pe date care pot fi restaurate în întregime (de exemplu când se lucrează pe un depozit de date pentru a calcula statistici sau pentru extragere de cunoștințe din date – data mining).
- ◆ Cerințele în acest caz pot fi mai mici pentru sistemele NoSQL.

F. Radulescu, Curs: Baze de date I

14

6. Lucrul cu masive de date

- ◆ În privința ultimelor două caracteristici – lucrul cu masive de date și eficiența – cerințele sunt mai mari în cazul sistemelor NoSQL decât în cazul celor tradiționale.
- ◆ De exemplu, în cazul rețelelor de socializare, volumul de date colectat în unitatea de timp este enorm.
- ◆ De aceea cerințele privind facilitățile de lucru cu volume mari de date sunt mai mari decât în cazul unui sistem tradițional.

F. Radulescu, Curs: Baze de date I

15

7. Eficiența

- ◆ Din cauza volumelor mari de date care trebuie prelucrate în timp util, cerințele privind eficiența (incluzând viteza de prelucrare) sunt mult mai mari decât la un sistem clasic.
- ◆ De aceea soluțiile includ de obicei operarea paralelă pe date.

F. Radulescu, Curs: Baze de date I

16

Deci

- ◆ Sistemele NoSQL au deci câteva caracteristici care le diferențiază de cele tradiționale:
1. O schemă flexibilă a datelor
 2. Sunt ușor și ieftin de instalat
 3. Scalabilitate – lucrul cu volume mari de date
 4. Consistența datelor este relaxată pentru a crește performanțele și disponibilitatea.

F. Radulescu, Curs: Baze de date I

17

Dezavantaje

- ◆ În unele cazuri nu există un limbaj declarativ de operare. Rezultă că operarea se poate face doar scriind programe de acces.
- ◆ Relaxarea consistenței nu le face o soluție bună pentru aplicații care au nevoie de garantarea consistenței, a serializabilității și a tranzacțiilor.

F. Radulescu, Curs: Baze de date I

18

Tipuri de sisteme NoSQL

◆ In [Widom 2] sunt enumerate patru categorii de sisteme NoSQL:

1. Framework-ul MapReduce
2. Stocare cheie-valoare
3. Stocare documente
4. Baze de date de grafuri

F. Radulescu, Curs: Baze de date 1

19

MapReduce

- ◆ Introdus de cei de la Google.
- ◆ In momentul de fata exista varianta open source Hadoop.
- ◆ In acest caz nu exista un model de date, acestea fiind stocate in fisiere.
- ◆ Exista doua sisteme de fisiere folosite, Google File System (GFS) pentru MapReduce (Google) si Hadoop Distributed File System (HDFS) pentru Hadoop.

F. Radulescu, Curs: Baze de date 1

20

Functii

- ◆ Framework-ul - care asigura o functionare distribuita - furnizeaza toate caracteristicile necesare pentru functionare, inclusiv toleranta la defecte si scalabilitatea.
- ◆ Pentru a il folosi utilizatorul trebuie sa scrie un set de functii necesare prelucrarii.

F. Radulescu, Curs: Baze de date 1

21

Functii

- ◆ Printre functiile care trebuie furnizate de utilizator exista doua care dau si numele de mai sus, si anume functia map() si functia reduce() .
- ◆ Pe langa acestea sunt si alte functii care in mod normal trebuie sa existe:
 - ◆ reader() este o functie care citeste datele din fisiere si le returneaza ca inregistrari
 - ◆ writer() ia inregistrările produse si le scrie in fisiere

F. Radulescu, Curs: Baze de date 1

22

Map

- ◆ Functia map() imparte o problema in subprobleme.
- ◆ Ea primeste un articol de date - o inregistrare furnizata de functia reader() - si returneaza un numar (≥ 0) de perechi (cheie, valoare).
- ◆ Cum prelucrarea este paralela, exista mai multe instante ale lui map() care lucreaza in paralel.

F. Radulescu, Curs: Baze de date 1

23

Map

- ◆ Aceste instante ale lui map() primesc date de la mai multe instante ale lui reader() care si ele lucreaza in paralel.
- ◆ Datele sunt citite de reader() din fisierele de date care reprezinta adevarata intrare in sistem.

F. Radulescu, Curs: Baze de date 1

24

Reduce

- ◆ Functia `reduce()` primește o cheie și o lista de valori asociate și returnează un număr de înregistrări (≥ 0).
- ◆ Iesirea lui `reduce()` este scrisă de `writer()` în fișierele de date de ieșire.
- ◆ Intrarea funcției `reduce()` este dată de ieșirile din instanțele funcției `map()`.

F. Radulescu. Curs: Baze de date 1

25

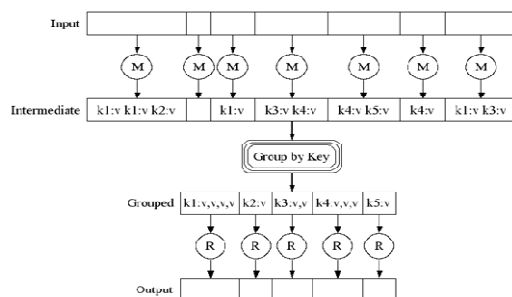
Reduce

- ◆ Aceste ieșiri sunt reunite de o funcție `combine()` care și ea trebuie să fie furnizată de utilizator.
- ◆ Funcția `combine` grupează perechile (cheie, valoare) după cheie, furnizând intrarea lui `reduce()` în formatul (cheie, lista de valori).
- ◆ Figurile următoare se găsesc în articolul [Dean, Ghemawat] publicat de Google.

F. Radulescu. Curs: Baze de date 1

26

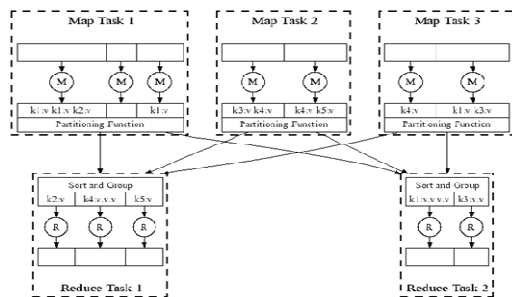
Executie



F. Radulescu. Curs: Baze de date 1

27

Executie paralela



F. Radulescu. Curs: Baze de date 1

28

Limbaje pentru MapReduce

- ◆ Pentru cei care nu doresc să facă munca de programare necesară în acest caz au fost dezvoltate și pachete care se pun peste Hadoop sau MapReduce:
 - ◆ Apache Hive este o infrastructură de tip data warehouse construită peste Hadoop pentru a furniza utilizatorului sumări, cereri (tip SQL) și analiza datelor.
 - ◆ Pig - seamănă cu algebra relatională
 - ◆ Dryad, etc

F. Radulescu. Curs: Baze de date 1

29

Tipuri de sisteme NoSQL

1. Framework-ul MapReduce
2. Stocare cheie-valoare
3. Stocare documente
4. Baze de date de grafuri

F. Radulescu. Curs: Baze de date 1

30

Stocarea cheie - valoare

- ◆ In acest caz se stocheaza perechi de forma <cheie, valoare>.
- ◆ Este un model de date foarte simplu dar care este util in multe cazuri.
- ◆ Operatiile principale sunt:
 - ◆ Insert(cheie, valoare) - inserarea unei noi perechi
 - ◆ Fetch(cheie) - incarcarea de valori dupa cheie
 - ◆ Update(cheie) - actualizare dupa cheie
 - ◆ Delete(cheie) - stergere dupa cheie

F. Radulescu. Curs: Baze de date I

31

Implementare

- ◆ Inregistrarile continand perechi sunt distribuite in noduri pe baza valorii cheii.
- ◆ In acest fel cautarea unei anumite chei se face doar in nodurile unde este posibil sa se gaseasca acea valoare
- ◆ Replicare: aceeasi pereche se gaseste in mai multe noduri (duplicate). Exista o consistenta intarziata - vezi mai jos.

F. Radulescu. Curs: Baze de date I

32

Implementare

- ◆ Fiecare operatie este o tranzactie.
- ◆ Consistenta intarziata: copiile replicate ale unei aceleiasi inregistrari nu sunt mereu identice ci doar la sfarsitul procesarii datelor (dupa 'convergenta').
- ◆ Eventual consistency este definita astfel in articolul De-mystifying "eventual consistency" in distributed systems publicat de Oracle (vezi si: <http://www.oracle.com/technetwork/products/nosql/db/documentation/consistency-explained-1659908.pdf>)

F. Radulescu. Curs: Baze de date I

33

Eventual consistency

- ◆ A distributed system maintains copies of its data on multiple machines in order to provide high availability and scalability.
- ◆ When an application makes a change to a data item on one machine, that change has to be propagated to the other replicas.
- ◆ Since the change propagation is not instantaneous, there's an interval of time during which some of the copies will have the most recent change, but others won't. In other words, the copies will be mutually inconsistent.
- ◆ However, the change will eventually (RO: in cele din urma) be propagated to all the copies, and hence the term "eventual consistency".
- ◆ The term eventual consistency is simply an acknowledgement that there is an unbounded delay in propagating a change made on one machine to all the other copies.
- ◆ Eventual consistency is not meaningful or relevant in centralized (single copy) systems since there's no need for propagation.

F. Radulescu. Curs: Baze de date I

34

Valori

- ◆ Valorile din perechile <cheie, valoare> nu sunt uniforme (ca tip) si unele pot fi structurate.
- ◆ Operatia FETCH este uneori extinsa, ea incarcand inregistrari care au cheia intr-o plaja de valori - deci nu doar pentru o singura valoare a cheii.
- ◆ Exemple de sisteme: GoogleBigTable, Amazon Dynamo, Hbase, etc.

F. Radulescu. Curs: Baze de date I

35

Tipuri de sisteme NoSQL

1. Framework-ul MapReduce
2. Stocare cheie-valoare
3. Stocare documente
4. Baze de date de grafuri

F. Radulescu. Curs: Baze de date I

36

Stocare documente

- ◆ Sistemele de stocare documente (document stores, magazine de documente) pot fi considerate tot sisteme pentru perechi <cheie, valoare>, valoarea fiind un document .
- ◆ Tipurile de documente care pot fi stocate difera in functie de sistem.
- ◆ Putem avea diverse formate: JSON (Java Script Object Notation), XML, alte formate semistructurate

F. Radulescu. Curs: Baze de date I

37

Exemplu de fisier JSON

```
{ "menu": {  
  "id": "file",  
  "value": "File",  
  "popup": {  
    "menuitem": [  
      { "value": "New", "onclick": "CreateNewDoc()" },  
      { "value": "Open", "onclick": "OpenDoc()" },  
      { "value": "Close", "onclick": "CloseDoc()" }  
    ]  
  }  
}}
```

F. Radulescu. Curs: Baze de date I

38

Acelasi fisier in XML

```
<menu id="file" value="File">  
  <popup>  
    <menuitem value="New" onclick="CreateNewDoc()" />  
    <menuitem value="Open" onclick="OpenDoc()" />  
    <menuitem value="Close" onclick="CloseDoc()" />  
  </popup>  
</menu>
```

(Sursa: <http://json.org/example.html>)

F. Radulescu. Curs: Baze de date I

39

Operatii

- ◆ Operatiile principale sunt ca si inainte Insert, Fetch, Update si Delete, operatii bazate pe cheie.
- ◆ Exista de asemenea Fetch pe baza continutului documentului.
- ◆ Exemple de sisteme:
 - ◆ MongoDB
 - ◆ CouchDB
 - ◆ SimpleDB, etc

F. Radulescu. Curs: Baze de date I

40

Tipuri de sisteme NoSQL

1. Framework-ul MapReduce
2. Stocare cheie-valoare
3. Stocare documente
4. Baze de date de grafuri

F. Radulescu. Curs: Baze de date I

41

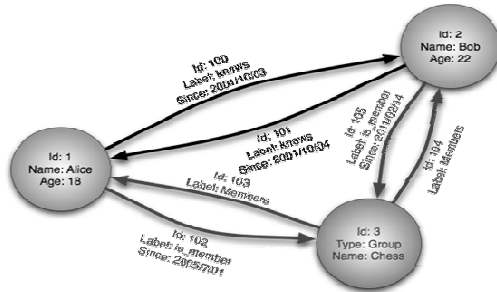
Baze de date de grafuri

- ◆ In acest caz obiectele sunt noduri si arce.
- ◆ Nodurile au proprietati (perechi <cheie, valoare>, de exemplu <ID, 3451>)
- ◆ Arcele pot avea etichete sau roluri.
- ◆ Un exemplu este urmatorul:
http://en.wikipedia.org/wiki/File:GraphDatabase_PropertyGraph.png

F. Radulescu. Curs: Baze de date I

42

Exemplu



F. Radulescu. Curs: Baze de date I

43

Bibliografie

- ◆ [Widom 1] Jennifer Widom – NoSQL System – Motivation, 2011
<http://www.youtube.com/watch?v=w0lPQZz7qjw>
- ◆ [Widom 2] Jennifer Widom – NoSQL System – Overview, 2011
http://www.youtube.com/watch?v=x-6Rsw_fQfE
- ◆ [Dean, Ghemawat] Jeff Dean, Sanjay Ghemawat, MapReduce: Simplified Data Processing on Large Clusters,
<http://research.google.com/archive/mapreduce-osdi04-slides/index.html>

F. Radulescu. Curs: Baze de date I

44

Sfarsitul ultimului capitolul

Bafta la examen!

F. Radulescu. Curs: Baze de date I

45