**2.ce este spark?**

Spark este un alt framework care face acelasi lucru ca si MapReduce. Dar il face mult mai repede.

**3.Cum a aparut?**

Matei Zaharia s-a nascut in Romania, dar apoi familia lui s-a mutat in Canada unde a absolvit unitatsitatea Waterloo. El a inceput sa lucreze la Spark cand inca era student, iar lucrarea lui de doctorat a facut posibila aparitia spark-ului.

Cand a fost intrebat cum i-a venit ideea de a incepe un asemenea proiect, el a spus ca a lucrat cu primii utilizatori importanti care foloseau Hadoop si a vazut tot felul de use-case-uri care nu se potriveau pentru MapReduce. O directie importanta a fost cea a Machine Learning-ului. Unde algoritmii sunt iterativi.

**4. 100 TB de date -> recorduri**

explic

**5. Spark vs MapReduce**

Cel mai mare avantaj este ca Spark incearca sa tina datele in memorie, in timp ce MapReduce dupa o serie de pasi scrie datele din memorie pe disk, apoi le citeste si continua procesarea datelor.

**6. Spark vs MapReduce**

Doar pastrand datele in memorie, viteza de procesare a Spark-ului este de 80 de ori mai mare decat cea de la MapReduce

7 **Spark vs MapReduce**

Este mult mai general si mai usor de folosit, pentru ca are mai multe metode si sunt mult mai intuitive

**8. Procesarea datelor**

Sa ne imaginam ca avem datele memorate ca o colectie sa spunem lista.  
Pe o masina, pe un singur fi de executie putem sa facem acest lucru in felul urmator:

foreach element din colectie, aplica functia asta

Daca am avea **N** fire de executie putem sa creem un wrapper care sa imparta lista in **N** parti si apoi fiecare fir de executie va procesa o parte din acea lista. Sunt limbaje de programare care au implementat astfel de colectii (ParArray, ParVector)

Cum putem sa procesam datele atunci cand sunt mai multe masini care memoreaza aceleasi date?

**9.RDD**

Procesarea in cluster presupune ca fiecare masina are cate o bucata din colectia mea, iar spark a creat o abstractie care sa simuleze lista. Cu alte cuvinte RDD-urile pot fi privite ca o simpla colectie, iar Spark-ul va face management-ul datelor in spate.

**10. Executia unui program in spark**

Programele in spark sunt de tipul master/slave

Executia unui program spark

1. Driver-ul ruleaza aplicatia Spark si creeaza SparkContext-ul de la inceput
2. SparkContextul se conecteaza la managerul clusterului (Mesos/ YARN) care aloca resursele
3. Spark primeste executorii din claster, acetia vor face computatia si vor stoca datele din aplicatie
4. Driver-ul trimite codul aplicatiei catre executori.
5. SparkContextul trimite task-uri executorilor si acestia ruleaza ce li se cere.

**11. Cum se creeaza un RDD**

**12. WordCount**

fara o actiune nu se executa nimic.

13. Laziness

14. Demo

* + sa scriu invers pentru ca filtrarea cu ERROR sa fie prima
  + creez la tabla DAG-ul
* spark-ul se opreste atunci cand a gasit 10 elemente care trec de filtru

15. poate o sa povestesc pe acelasi exemplu

**16. O noua abstractie - dataFrame**

As fi putut sa scriu ***un nou API*** , dar de fapt, API-ul folosit pentru RDD-uri a ramas aproape acelasi, cu foarte mici modificari, iar API-ul de la RDD-ul este aproape la fel ca operatiile pe care le poti face pe orice colectie in scala. Ceea ce e un avantaj. Exista continuitate.

19.

Prin diverse mecanisme se genereaza date, care sunt modelate cu ajutorul Machine Learning-ului, prin Big data putem pune aceste informatii intr-un context si sa formam cunostinte, iar aceste cunosinte, daca sunt manipulate asa cum trebuie pot crea aplicatii la care inca nici nu va ganditi ca e posibil sa fie.

Dar totul pleaca de la colectia de date, tehnologiile cu care pot fi procesate aceste date (Spark) si modalitatile de la procesa datele brute (Machine Learning)

20.

Google brain team au facut urmatorul experiment: Au creat 3 retele neuronale: Alice, Bob si Eve. Alice si Bob au schimbat initial niste numere random intre ei (o posibila cheie)  
  
Alice trebuie sa trimita un mesaj criptat lui Bob

Bob trebuia sa de decripteze mesajul

Eve trebuia sa incerce sa intercepteze mesajul si sa-l descifreze.

Dupa 15.000 de iteratii Bob putea sa decripteze mesajul de la Alice, iar Eve reusea sa descrifreze doar 8 din cei 16 biti.

Dar inca nu a fost pus in practica pentru sistemul este prea complex pentru a fi inteles si oamenii se tem sa nu aiba vulnerabilitati.