

1. Bevezetés

A big data napjainkban az egyik vezető címke az informatikai terminológiák körében, különböző címekkel ellátva, mint: *Sikerhez és boldogsághoz vezet a big data* [?] vagy *Új korszak kezdődött a tudományban.* [?] De mit is jelent pontosan? Nincs explicit kimondott meghatározás a fogalomra, de Doug Laney 2001-es definíciója egy jó indulópontnak tekinthető: az adatok nagy mennyiségben (volume), gyorsan(velocity) és különböző formátumban(variety) jelennek meg (3V's) [?].Azonban, ma már kiegészíthetjük ezt a fogalmat még 2V-vel: bizonyosság(veracity) és érték(value). [?] Az adatmennyiség amit előállítunk exponenciálisan növekszik olyan szintre, aminek tárolását, menedzselését és elemzését már nem tudjuk megoldani a saját, lokális erőforrásainkon belül az eddig megszokott adatelemzési eszközökkel, mint például Microsoft Excel, vagy különböző relációs adatbázis technológiák által. Becslések [?] szerint az adatok mennyisége kétévente duplázódik, így 2020-ra az összekészben forgó adatmennyiség elérheti a 44 zetabájtnyi (vagy 44 trillió gigabájtnyi) mennyiséget.

A "big data" lehetőséget biztosít arra, hogy ezeket az adatokat ne csak tároljuk, hanem új módokon tanuljunk belőle, értéket állítsunk elő, többet megtudjunk ügyfeleinkről, a saját üzleti folyamatainkról, ami versenyelőnyhöz vezethet. E mellett az áttörő kutatások számát is megnöveli azáltal, hogy rejtett összefüggéseket mutat meg. [?]

A cloud computing, és új technológiák megszületése és az, hogy a fizikai világ egyre jobban áttérrelődik az online térbe, új nehézségeket állít elő mind az adatokat kiszolgáló, mind az adatokat elemző infrastruktúrák számára. Ezek a problémák komoly gondot jelentek az informatikai iparnak, mivel érintik az fizikai manifesztációt (hardver), mind az ezt vezérlő és feldolgozó réteget (szoftver és algoritmus). Ezek a problémák, [?] –amelyek a tradicionális adattárház technológiákra jellemzőek– többek között származhatnak a hiba-tolerancia hiányából, a sokféle adatfajtából, a párhuzamosság hiányából, mely azt eredményezi, hogy a mai technológia fejlettség (és a központi számítási egységek fizikailag limitáltsága miatt) nem lesz megfelelő számítási teljesítmény a megnövekedett adatmennyiség menedzselésére.

2. A dolgozat célja

A technológia fejlődése és a számítási teljesítmény megnövekedése hozta létre azt az üzleti igényt [?], hogy egyre gyorsabban, egyre nagyobb adatmennyiség feldolgozása történjen meg. Ilyen igény például: csalás felderítés [?],

"dolgozók" internete (IoT) [?] vagy alkalmazás monitoring [?]. Ez az adatfeldolgozási sebesség olyan szintre eljutott, hogy közel valós időben, az adat keletkezése után megtörténhet ennek feldolgozása. Ilyen gyorsaságú adatfeldolgozásra csak elosztott rendszerek segítségével vagyunk képesek, [?] amelyek felépítésükből fakadóan sok lehetőség és költség jellemző, amelyeket a későbbiekben fogok kifejteni. A dolgozatomban használt Apache Flink (mely az Apache Foundation egyik legújabb és legmodernebb terméke) platform közel 40 millió elem feldolgozására képes egy 40 magos architektúrán másodpercenként. [?].

Ahhoz, hogy ezt az adatmennyiséget ki tudjuk elemezni és ajánlásokat tudjunk adni, gépi tanulásra van szükségünk. A gépi tanulás az informatikának és a matematikának egy olyan ága, amely az adatok folyamatos betáplálása során új ismereteket szolgáltat, megpróbál előrejelzéseket adni anélkül, hogy explicit módon be lenne erre programozva. [?]. Az választott módszer a stochastic gradient descent (SGD, sztochasztikus gradiens ajánlás) [?], amely egy olyan egyszerűsítési illetve optimalizációs eljárás, ahol adott célfüggvény gradiensét folyamatosan, iteratív módon számoljuk ki. Céлом az, hogy megtervezzem Apache Flink alatt az SGD algoritmust, összehasonlítom a teljesítményét a már implementált algoritmusokkal és megkezdjem a szükséges módosítások implementálását.

3. Gépi tanulás

4. Ajánlórendszerek

5. Batch

6. Lambda-architektúra

7. Streaming

8. Határos-határtalan

9. Time-agonistic

- Event time
- Processing time

10. Flink(Spark/Storm/Mapreduce)

11. Ajánlórendszerek

- Collaborative filtering
- Content based

12. ALS

13. DSG

14. Java, Scala

Max 1 oldal, miért ez lett

15. Dcg,nDCG