Zbiranje in obdelava velikih količin podatkov

Borut Rožac

TelekomSlovenije

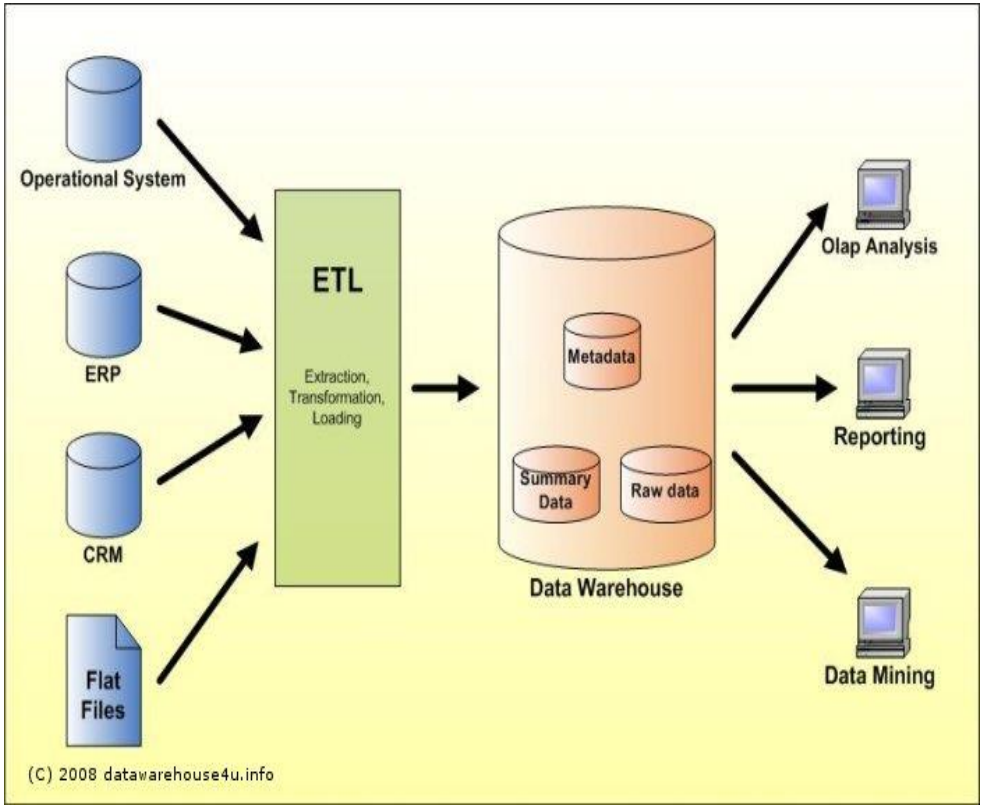
Vsebina

- Analize podatkov v podjetjih
- Običajne relacijske baze
- NOSQL in HADOOP
- Orodja za analitiko
- Primer zajemanja podatkovnega toka (straming)
- Uporaba Java zbirk za poizvedbe
- Analiza geografskih podatkov

Analiza podatkov v podjetjih

Zelo dobro uveljavljeno področje

- Računovodstvo
- Finance
- Kontroling
- Uveljavljeni postopki in rešitve
  - EXCEL
  - SQL produkti
  - ETL + DWH
  - OpenSource
    - BIRT
    - R, Python,…
  - Profesionalna orodja
    - Tableau, SAS, SPSS,…
- Novi izzivi
  - **Neskončne nove želje**
  - **Big Data**



Običajne relacijske baze

Zelo dolgo na sceni (dobrih 20 let)

- Podatki so v osnovi v tabelah
- Plačljive
  - **ORACLE, MSSQL,DB2,..**
- Brezplačne
  - **MYSQL, PGSQL, POSTGIS, H2,…**
- Specializirane (namenske, predvsem analitika)
  - **Netezza, Vertica, EXASOL,…**
- Namenjene za `poslovno informatiko`
  - **Integriteta podatkov (ACID)**
  - **Dostopnost gonilnikov (ODBC, JDBC,…)**
  - **Uveljavljena uporaba**
  - **Uveljavljeno upravljanje in vzdrževanje**
- Dobre za transakcije (OLTP)
  - **Pomanjkljive za novejše pristope, predvsem oblike podatkov**

Namenske SQL baze

Namenjene ožji uporabi

- V osnovi namenjene analitiki
- Potreba po agregatih in povezovanju (**GROUP BY in JOIN**)
- Žrtvujejo del funkcionalnosti na račun zmogljivosti. (FK-ji ni indeksov)
- V osnovi gre za column-based ali hibridne baze.
- Uporabljajo bodisi FPGA ali RAM (dobro, da ga je čim več)
- ANSI SQL z omejitvami
- Integracija z R, Javo, LUA, Pythonom direkt
- Običajni gonilniki za večino programskih jezikov
- V BI orodjih podprte direkt
- Nekatere podpirajo ML preko SQL jezika (Vertica, Oracle tudi)
- EXASOL, NETEZZA (IBM), VERTCA (HP), Oracle EXADATA (kombinacija)
- Določene funkcionalnosti podprte v običajnih bazah

NoSQL in HADOOP

HADOOP

- Distribuirana zbirka podatkov in zbirka produktov
  - Hadoop, YARN, Hive, HBase, Spark, Storm, Sqoop
- Namenjena v osnovi razširjeni obdelavi podatkov
  - Shranjevanje različnih tipov (strukturirani podatki)
  - Običajne SQL baze niso primerne (horizontalna skalabilnost)
- Dobra podpora za razvoj in uporabo
  - Dostopne knjižnice za večino programskih jezikov
  - Dostop mogoč iz večine komercialnih orodij (SAS, Tableau)
- Integrirano v komercialne baze (Oracle, Exasol, Vertica)

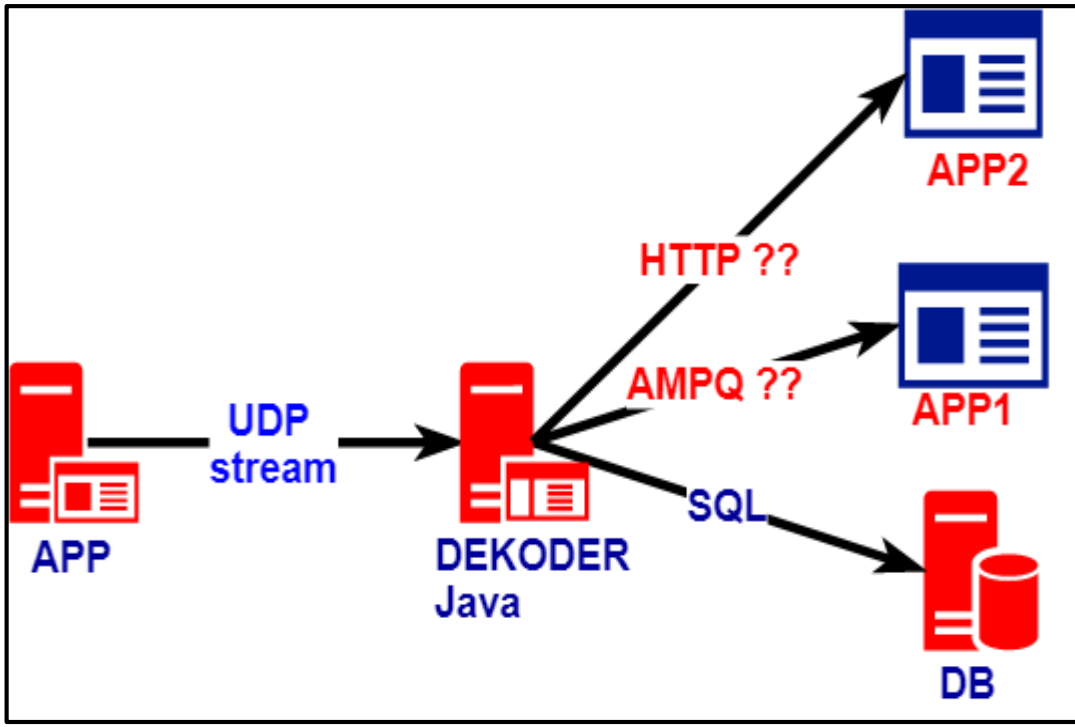
NoSQL

- Podatki niso v osnovi tabelarični
- Na račun hitrosti se žrtvuje konsistenca
- Ključ - vrednost
- Dokumenti
  - Twitti, logi, komentarji
- Poizvedbe večinoma podprte kot SQL v osnovi pa ne!
- MongoDB, CouchDB, neo4j,
- Podpora tudi za ACID

Spark vs Java Code

Problem: Zbiranje streama oktetov (byte[]) + dekodiranje + shranjevanje + real time obdelava

- Spark streaming modul (naravna izbira)
- Dobra dokumentacija
- Java API za Spark (Maven)
- Stream Receiver
  - Basic (datoteke, sockets)
  - Kafka, Flume, Kinesis
- Problem UDP stream
- Rešitev:
  - Java UDP sockets
  - ConcurrentLinkedQueue
  - ScheduledExecutorService
- Še vedno potreben de-koder



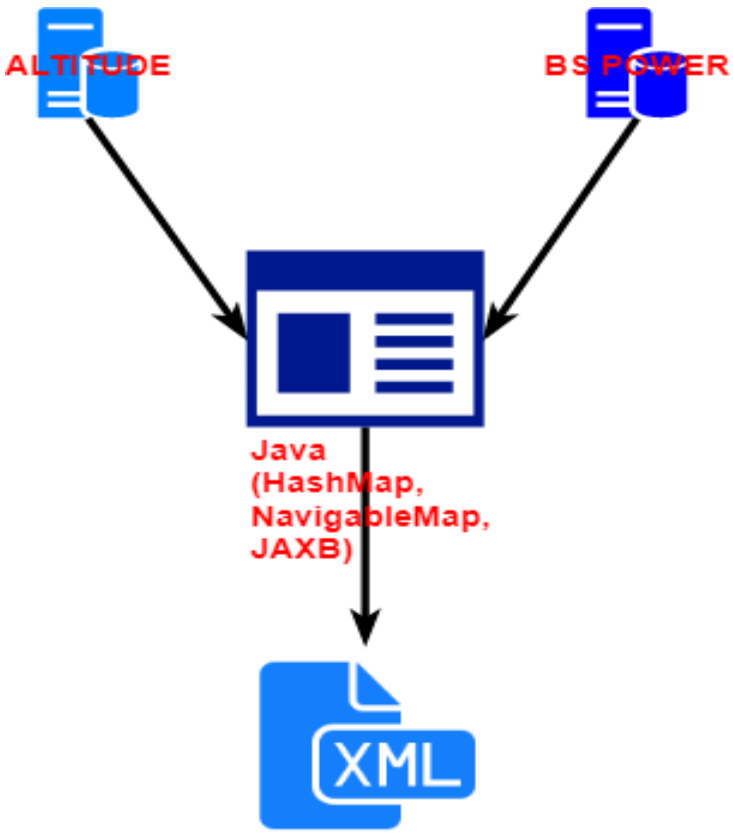
Spark vs Java Code (1)

- Prepustnost 1000~1500 paketov/s
- Praznjenje vrste na 15 sekund.
- Dekodiranje in shranjevanje v ločenem threadu.
- Tekoče delovanje ene leto.
- Testirano s **pcap4j** knjižnico.
- Slabosti
  - **Nobenih kontrol, vklop, izklop, restart (dodatna implementacija)**
  - **Potrebno dobro poznavanje posameznih knjižnic**
  - **Ni distribuirano**
  - **Potrebna integracija z drugimi sistemi (nadzor)**

```
ConcurrentLinkedQueue<byte[]> cclq = new ConcurrentLinkedQueue<>();
ScheduledExecutorService ses = Executors.newScheduledThreadPool(2);
Decoder dec = new Decoder(cclq);
ses.scheduleAtFixedRate(dec, 0, 15, TimeUnit.SECONDS);
try(DatagramSocket socket = new DatagramSocket(udpPort)){
    byte[] buffer = new byte[bufferLength];
    DatagramPacket dp = new DatagramPacket(buffer, buffer.length);
    for(;;){
        socket.receive(dp);
        byte[] data = dp.getData();
        cclq.add(data);
    }
}catch (Exception e) {
}
```

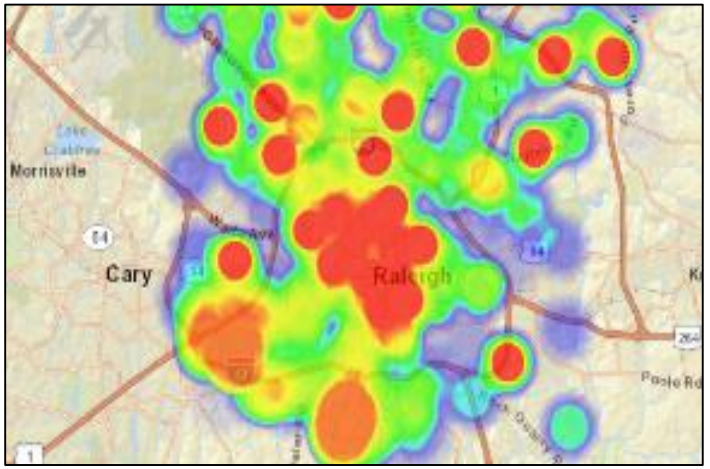
HashMap in NavigableMap

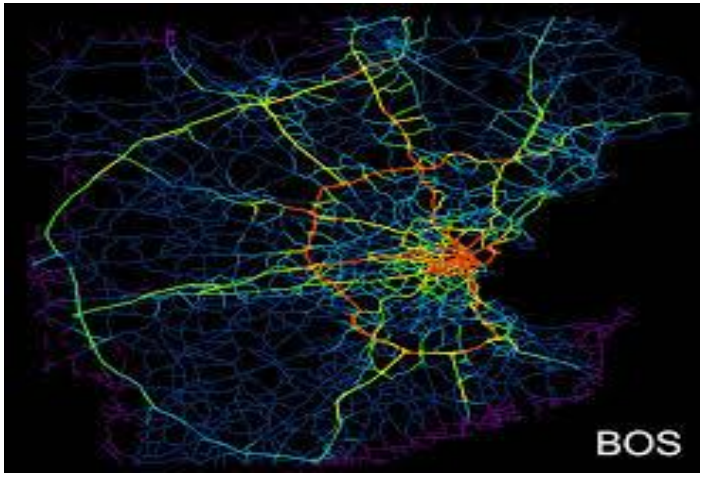
- **Problem zbiranje velikih količin podatkov iz dveh različnih baz**
  - Moči baznih postaj v prvi bazi
  - Nadmorske višine (geodetska uprava .shp)
  - Izdelati .xml dokument za določen sistem
- Poizvedbe za nadmorsko višino.
- Možnost kopiranja podatkov med bazama
  - **Možnosti sprememb v bazah!**
- **Poizvedbe na DB 10~100 ms.**
- Rešitev v obdelavi manjših enot (občina ali poštna št.)
- HashMap<Koordinata, Višina>
- **Poizvedbe <1ms (lokalno izvajanje).**
- Implementacija hashCode() in hashMap() metod za razred Koordinata
- **NavigableMap<Object,Object>** pretvorba intervalov



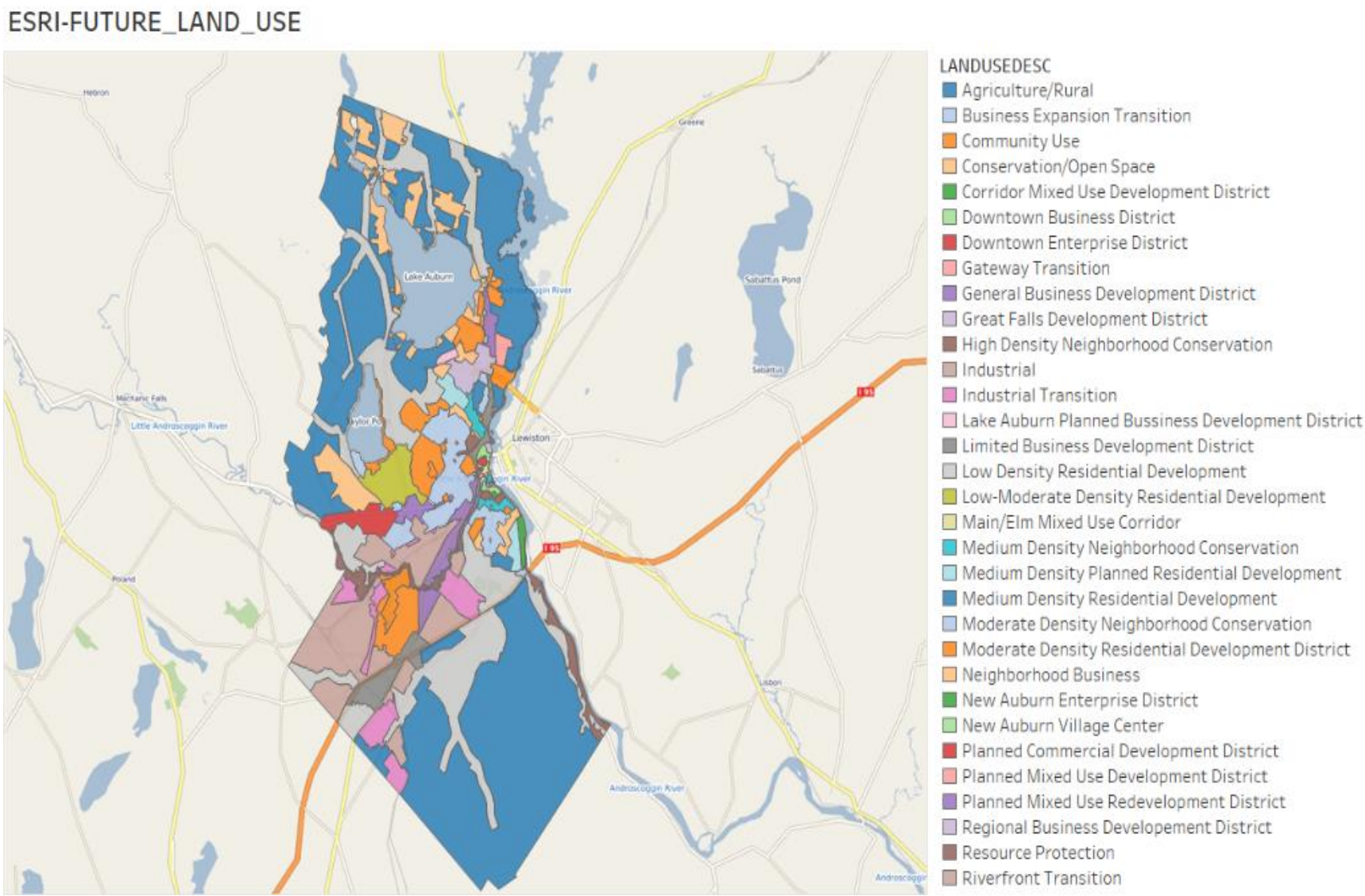
Analiza geografskih podatkov

- **Zelo pomembni pri analizah (IoT-lokacija, ostalo povezano z lokacijo)**
- OGS (Open Geospatial Consortium)
- Definirane podatki in metode za SQL in ostale knjižnice (R, java, Python)
- Veliko primerov in dobra dokumentacija
- Funkcionalnosti implementirane v večini baz SQL in NoSQL
  - PostGis, MySQL
  - Oracle, MSSQL, EXASOL,…
- Orodja
  - **QGIS (Brezplačen, veliko vtičnikov)**
  - **ArcGis (ESRI, brezplačno, knjižnice)**
  - **Analitska orodja (Tableau, SAS, EXCEL,..)**
- Datoteke
  - GeoJSON, KML, .shp, .mif,
- **! Koordinatni sistemi (EPSG 3912 si)**





Geo podatki (Namembnost zemljišč Lewiston MA.) vir. ESRI



Pregled orodij za analizo podatkov

- **R**
  - Prednosti: veliko knjižnic, IDE, široka baza znanja, pogojno brezplačen
  - Slabosti: Ni primeren za masovno produkcijo, potrebno min. dev. znanje
- **Tableau**
  - Prednosti: grafično okolje, intuitiven, lepe vizualizacije, gonilniki, podatkovni viri
  - Slabosti: plačljiv, ne da se narediti ravno vsega
- **Python**
  - Prednosti: razširjenost, dobra podpora (posebej ML in big data), izpodriva R
  - Slabosti: težek prehod iz drugih PJ, potrebno min. dev znanje
- **Ostala orodja**
  - **Vsako skuša podpreti določen nabor funkcionalnosti**
  - **Dobro razmisliti za kaj se bo orodje uporabilo**
  - **Kdo so ključni uporabniki**
  - **Učna krivulja**
  - **Podpora**
  - **Kako se da orodje in produkte integrirati z ostalimi sistemi**

Hvala!

Telekom Slovenije, d.d.
Cigaletova 15
1000 Ljubljana

www.telekom.si
T: 041 700 700 ali 080 8000
E: info@telekom.si

TelekomSlovenije