

Univerza v Ljubljani
Fakulteta za računalništvo
in informatiko

13.5.
2016

Predmet:
Podatkovne baze

Modul:
Kratek pregled NoSQL paradigme

Gradivo:
v.2015

Univerza v Ljubljani
Fakulteta za računalništvo
in informatiko

Vsebina

- BigData
- Relacijske vs NoSQL podatkovne baze
- Lastnosti in tehnologije porazdeljenih baz
 - Fragmentacija in replikacija
 - ACID lastnosti
 - CAP izrek
 - Postopna konsistentnost
 - Model Map Reduce
- Vrste noSQL podatkovnih baz in primerjava
 - Key-Value, Wide Column, Documet, Graph

Predmet: PB, Modul: Kratek pregled NoSQL paradigme, Gradivo: v.2015, ©UL FRI, Laboratorij za podatkovne tehnologije

2

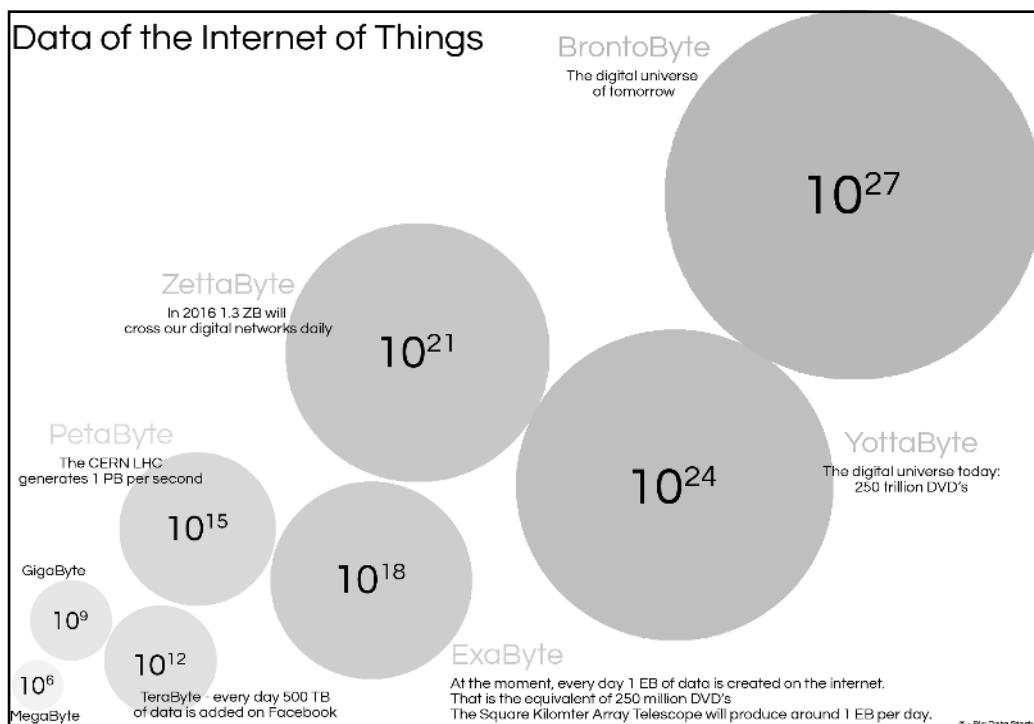
BigData...področja

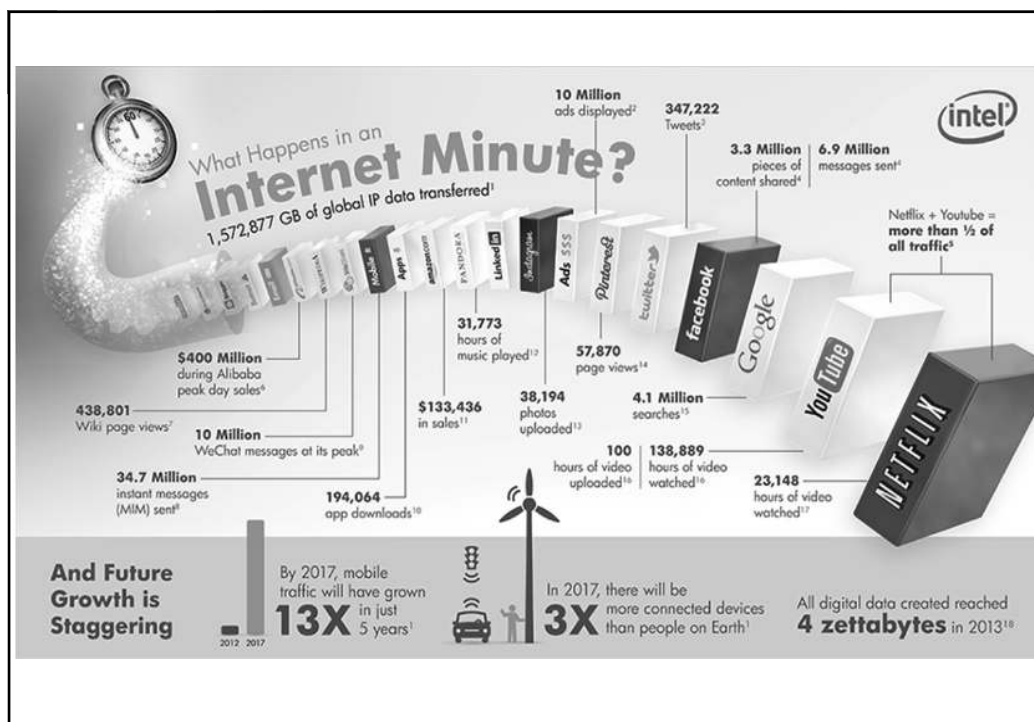
- Številni primeri, kjer imamo opravka z ogromnimi količinami podatkov...
 - Senzorska omrežja,
 - Družbena omrežja,
 - Spletni dokumenti
 - CDR
 - Astronomija
 - Genomika
 - Vojaški nadzor
 - Video arhivi
 - ...



Predmet: PB, Modul: Kratki pregled NoSQL paradigme, Gradivo: v.2015, ©UL FRI, Laboratorij za podatkovne tehnologije

3





Univerza v Ljubljani
Vakulteta za računalništvo in informatiko

BigData...število uporabnikov

- Posamezne spletne aplikacije uporabljajo milijoni uporabnikov... primer: družbeni mediji.

1,415 milijarde (2015)

- Pričakujemo 100% razpoložljivost in zanesljivost...

Predmet: PB, Modul: Kratki pregled NoSQL paradigme, Gradivo: v.2015, ©UL FRI, Laboratorij za podatkovne tehnologije

BigData...razširljiovost

- Nujna lastnost: razširljivost (scalability).
- Razširljiva aplikacija:
 - če imamo dovolj strežnikov, lahko zagotovimo fiksne odzivne časa ne glede na količino podatkov...
 - Idealno: linearna odvisnost med obsegom dela, časom za procesiranje ter številom strežnikov (vozlišča).
- Razširljivost po podatkih
- Razširljivost po številu uporabnikov

BigData...primer Amazon

- Primer razširljive spletne aplikacije: Amazon.
 - Začetek 1994 – spletna trgovina s knjigami
 - 2009: 70.000.000 obiskovalcev mesečno
 - Začetna arhitektura: monolitna aplikacija, spletni strežnik s poslovno logiko, en podatkovni strežnik.
 - 2011: uvedba SOA arhitekture



Relacijske vs NoSQL baze...

- Relacijske baze včasih uporabne za vsak problem...
- Danes problem porazdeljenost (transakcije, stične operacije...)
- Pojav nerelacijski baz z omejenim naborom funkcionalnosti...
NoSQL
- NoSQL paradigma: ne obstaja enotna tehnologija za obvladovanje podatkov, ki bi bila primerna za vse probleme!

Relacijske vs NoSQL baze...

- Relacijske baze
 - Zagotavljajo zanesljiv način izvajanja transakcij (ACID: Atomicity, Consistency, Isolation, Durability)
 - Standardizirani poizvedovalni jeziki
 - Podatkovna shema
 - Distribucija redko

Relacijske vs NoSQL baze

- NoSQL
 - Distribucija prek mreže poceni računalnikov
 - Zagotovljena varnost pred odpovedjo (Fault Tolerant).
 - ACID le delno zagotovljen.
 - Navadno brez podatkovne sheme
 - Optimizirane za spletne storitve
 - Lastni poizvedovalni jeziki

Lastnosti in tehnologije porazdeljeni baz

- Fragmentacija in replikacija
- ACID lastnosti
- CAP izrek
- Postopna konsistentnost
- Model MapReduce

Replikacija in fragmentacija

- Tri dimenzije razširljivosti podatkovnih baz:
 - Po številu bralnih operacij
 - Po številu pisalnih operacij
 - Po velikosti podatkovne baze
- Razširljivost zagotavljamo s pomočjo replikacije in fragmentacije (sharding). Navadno s kombinacijo obojega.

Replikacija...

- Zapis podatkovnih enot na več kot eno vozlišče.
- Učinkovitejše branje (load balancer).
- Večja zanesljivost (odpornost proti odpovedi posameznega vozlišča)
- Replikacija na različne podatkovne centre v izogib katastrofalnim nesrečam...
- Replikacija lahko delna ali popolna ("povsod vse")

Replikacija

- Negativen efekt na pisalne operacije... vse replike morajo biti zapisane.
- Možnosti:
 - Pisalna operacija izvedena na vseh vozliščih z repliko. Uspešna, ko potrjena na vseh vozliščih.
 - Pisalna operacija izvedena na enem ali nekaj vozliščih. Kasneje vzporedno asinhrono posredovana še drugim vozliščem.

Fragmentacija

- Fragmentacija (Sharding): razbitje podatkov na fragmente. Fragmenti zapisani na posameznih vozliščih.
- Vozlišča poljubno dodajamo... povečujemo kapaciteto ter učinkovitost branja in pisanja.
- Negativna plat – kompleksnost JOIN operacij. Navadno nepodprte v porazdeljenih bazah.

ACID lastnosti

- **ACID:**
 - Atomarnost (**A**tomicity)
 - Konsistentnost (**C**onsistency)
 - Izolacija (**I**solation)
 - Trajnost (**D**urability)
- Relacijske baze koristijo vnaprejšen zapis v dnevnik (write-ahead log) za **D** ter centralno zaklepanje (central lock) za **ACI**.
- V porazdeljenih bazah centralno zaklepanje ozko grlo. NoSQL uporabljajo poseben pristop.

Izrek CAP

- V porazdeljeni bazi lahko zagotovimo le dve od treh lastnosti:
 - Razpoložljivost (availability)
 - Konsistentnost (consistency)
 - Odpornost proti odpovedim posameznih particij (partition tolerance).

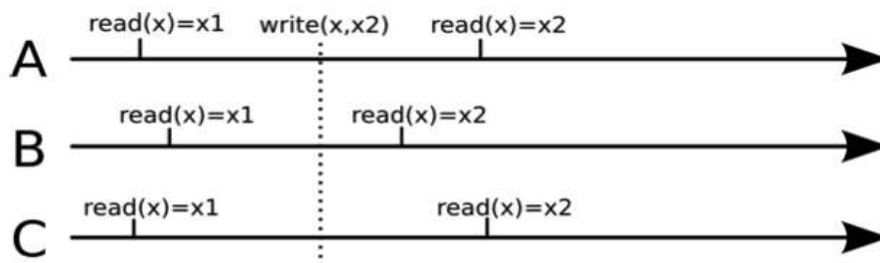


Konsistentnost...vrste

- Stroga konsistenost: vse verzije posamezne podatkovne enote so enake.
- Postopna konsistentnost: vse verzije posamezne podatkovne enote bodo sčasoma enake.
- Primer:
 - A, B, C: nepovezani procesi, ki berejo ali pišejo v bazo
 - x: opazovana podatkovna enota
 - x1, x2, x3: različne vrednosti podatkovne enote x

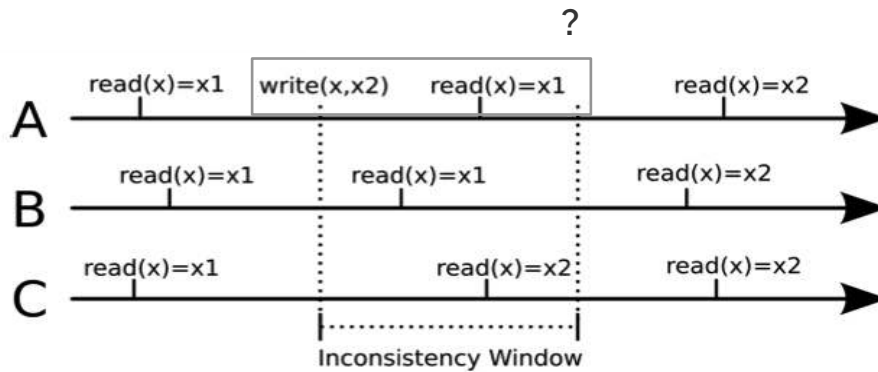
Konsistentnost...stroga

- Stroga konsistenost (Strict Consistency)



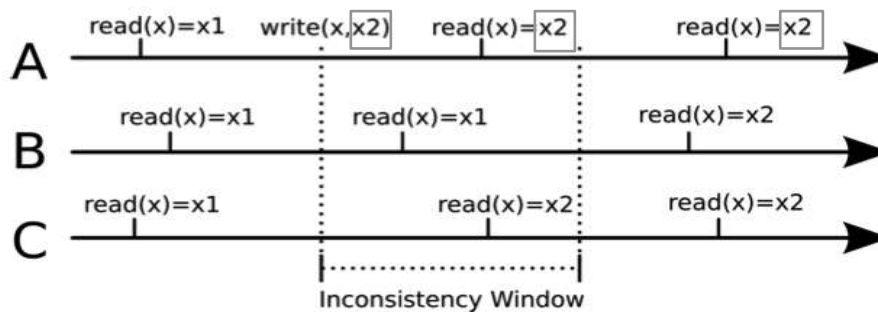
Konsistentnost...postopna...

- Postopna konsistenost (Eventual Consistency)



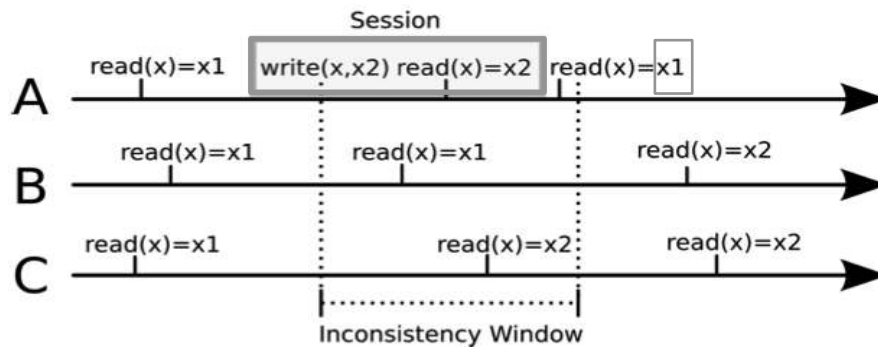
Konsistentnost...postopna

- Postopna konsistenost “beri-svoje-podatke” (Read-Your-Own-Writes Consistency)



Konsistentnost...seje

- Konsistenost seje (Session Consistency)

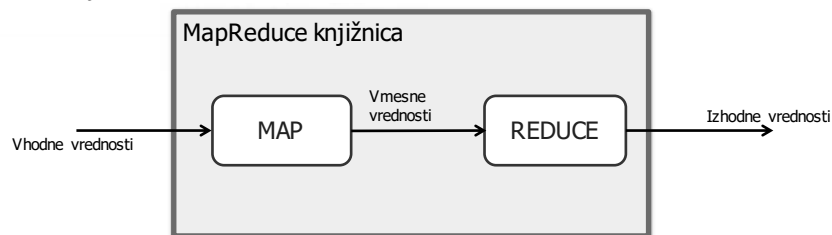


MapReduce...programski model...

- MapReduce je programski model/ogrodje za porazdeljeno računanje
- Razvil Google
- MapReduce ogrodje poskrbi za distribucijo po n strežnikih, za sinhronizacijo in paralelizacijo...
- Primeren za obdelavo velikih količin podatkov, ki jih ni moč obdelati na enem strežniku...

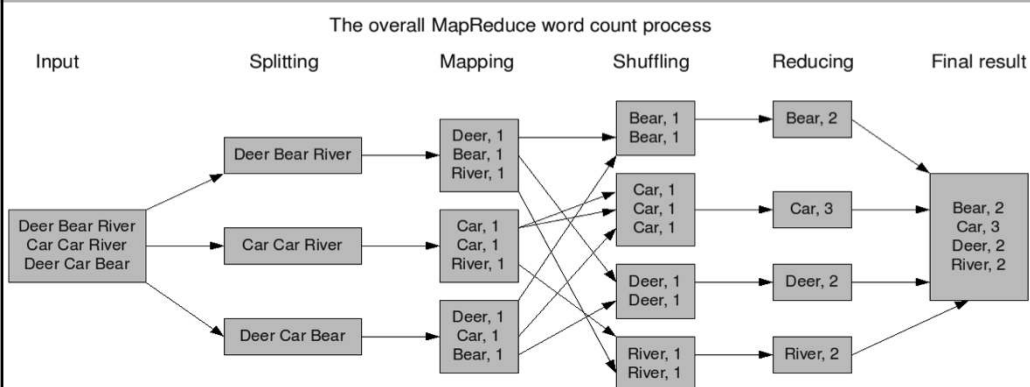
MapReduce...programski model

- MapReduce programiranje: definiramo MAP funkcijo in REDUCE funkcijo



- `map(key, value) → list(ikey, ivalue)`
- `reduce(ikey, list(ivalue)) → list(fvalue)`

Primer uporabe MapReduce – štetje besed



Primer povzet po: <http://blog.team.nl/2009/08/04/introduction-to-hadoop/>

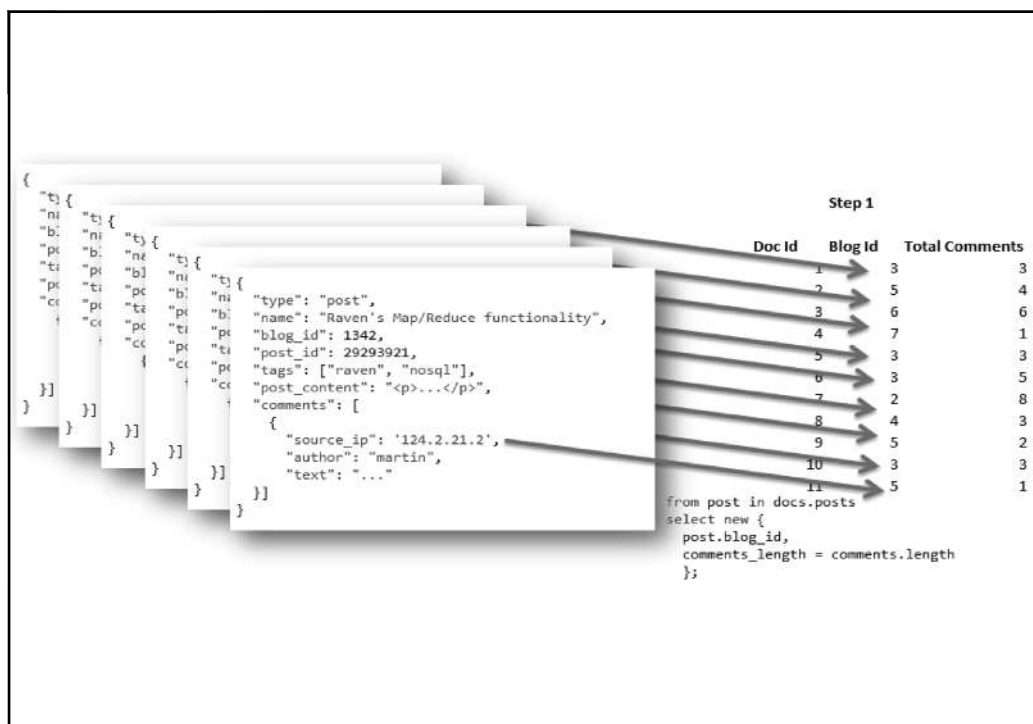


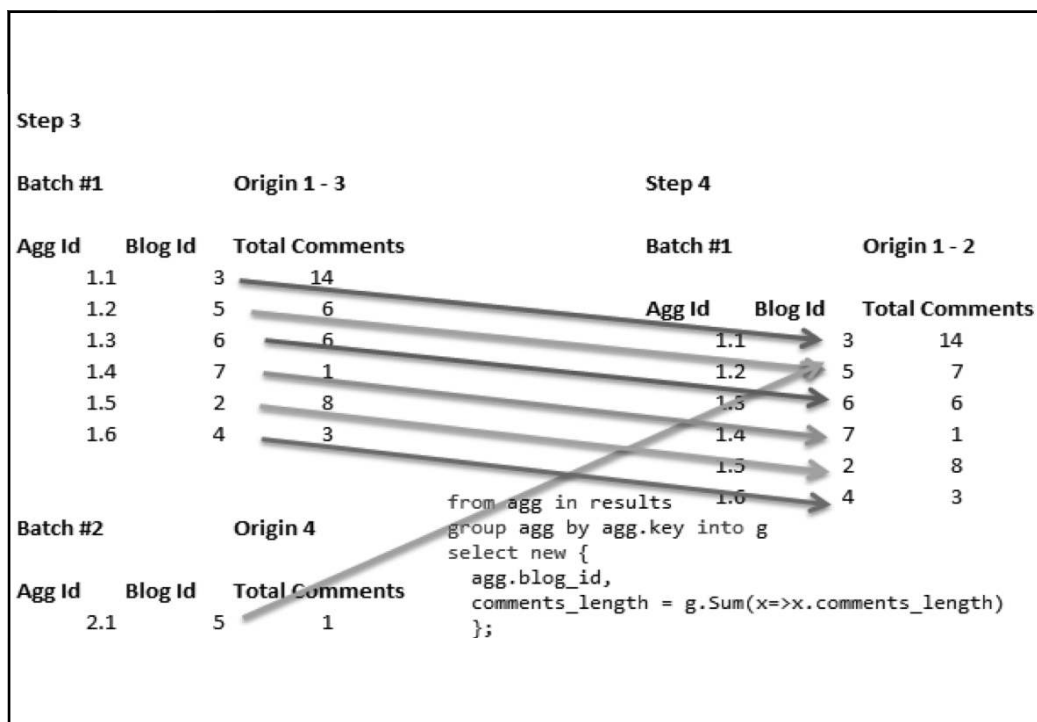
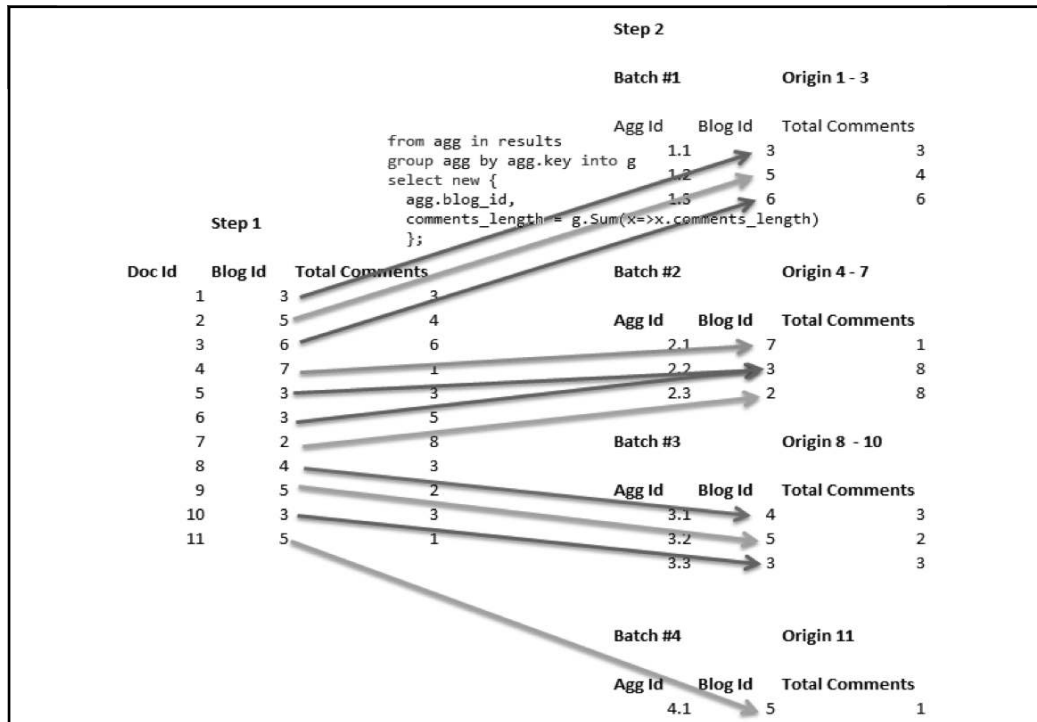
Primer uporabe MapReduce - štetje komentarjev

- Primer: štetje komentarjev v blogih

```
{
  "type": "post",
  "name": "Raven's Map/Reduce functionality",
  "blog_id": 1342,
  "post_id": 29293921,
  "tags": ["raven", "nosql"],
  "post_content": "<p>...</p>",
  "comments": [
    {
      "source_ip": '124.2.21.2',
      "author": "martin",
      "text": "..."
    }
  ]
}
```

Primer povzet po: <http://ayende.com/blog/4435/map-reduce-a-visual-explanation>





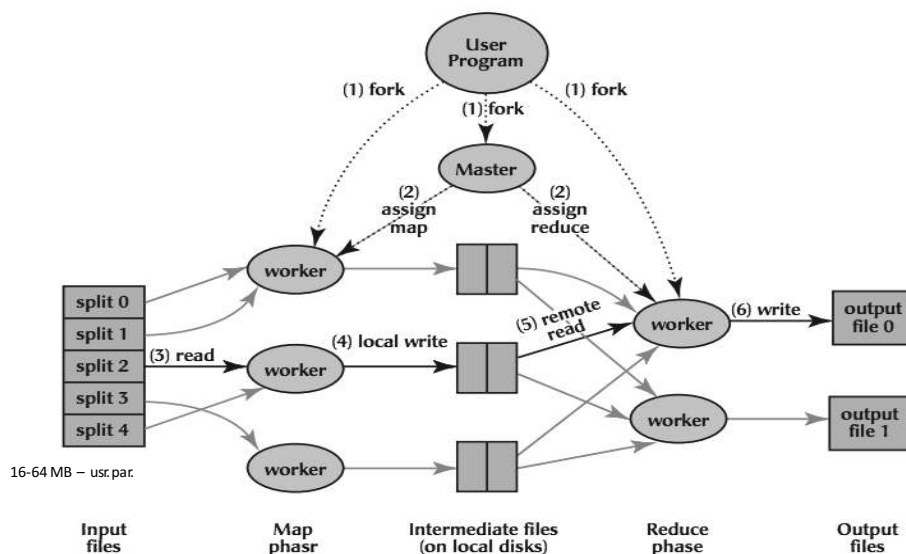
MapReduce...implementacije

- Številne različne implementacije
 - Večprocesorski strežniki z NUMA spominom
 - Računalniški gridi
 - ...
- Primer Google:
 - gruča več tisoč računalnikov:
 - 4-8 GB RAM
 - Dva X86 procesorja
 - 1 GB/s povezava
 - Lokalno priključeni diski,
 - Porazdeljen datotečni sistem GFS z replikacijo
 - Poseben sistem za razvrščanje opravil

Predmet: PB, Modul: Kratek pregled NoSQL paradigme, Gradivo: v.2015, ©UL FRI, Laboratorij za podatkovne tehnologije

31

MapReduce...Googlova implementacija



MapReduce...Hadoop



- Apache, odprto-kodni projekt. Vključuje več modulov:
 - Hadoop Common: skupne knjižnice, ki podpirajo druge Hadoop module.
 - Hadoop Distributed File System (HDFS™): porazdeljen datotečni sistem, ki zagotavlja visoko propustnost in dostop do podatkov.
 - Hadoop YARN: ogrodje za razvrščanje opravil in upravljanje z viri v računalniški gruči.
 - Hadoop MapReduce: na YARN-u osnovan sistem za paralelno procesiranje velikih podatkovnih zbirk.

NoSQL podatkovne baze



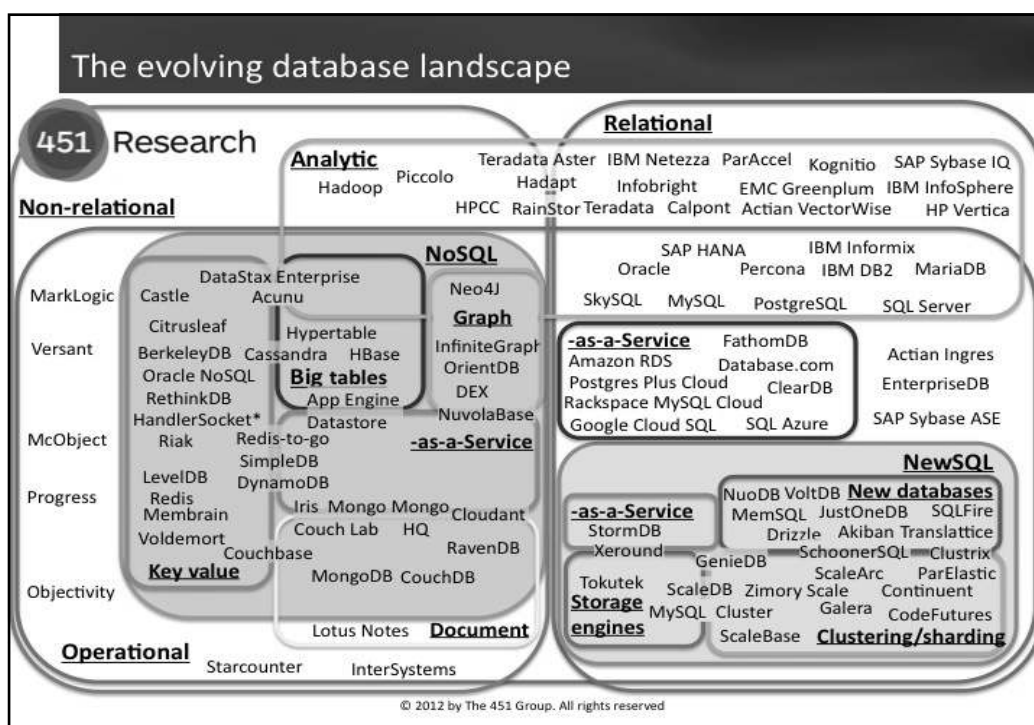
NoSQL...umestitev...

- Številni načini umestitve NoSQL
- Trenutni trendi:
 - NoSQL PB za razširljivost na porazdeljenih arhitekturah, podatki brez sheme...
 - NewSQL PB za porazdeljene arhitekture ali za povečanje učinkovitosti brez potrebe po horizontalni razširljivosti...
 - Data grid/cache produkti, razviti za hranjenje podatkov v hitrem spominu – za povečanje učinkovitosti...

Vir: <http://cdi-mdm.blogspot.com/2011/07/nosql-newsql-and-mdm.html>

Predmet: PB, Modul: Kratke pregled NoSQL paradigme, Gradivo: v.2015, ©UL FRI, Laboratorij za podatkovne tehnologije

35



NoSQL...vrste

- Key Value Store
 - Azure Table Storage, MEMBASE, GenieDB, Tokyo Cabinet / Tyrant, MemcachedDB, Amazon Dynamo, Voldemort
- Wide Column Store
 - Hadoop, Cassandra, Cloudata, Cloudera, Amazon SimpleDB
- Document Store
 - MongoDB, CouchDB, Citrusleaf
- Graph Databases
 - Neo4J, Infinite Graph, Bigdata

Podroben seznam: <http://nosql-database.org/>

Predmet: PB, Modul: Kratek pregled NoSQL paradigme, Gradivo: v.2015, ©UL FRI, Laboratorij za podatkovne tehnologije

37

NoSQL...Key-Value...

- Primer: facebook uporabniki

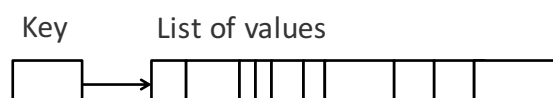
ID	First Name	Last Name
1	Samantha	Jones
2	James	Bond
3	James	Kirk

Key:	ID: sj	First Name: Sam
------	--------	-----------------

Key:	Email: jb@gmail.com	Location: London	Age: 37
------	---------------------	------------------	---------

Key:	Facebook ID: jkirk	Password: xxx	Name: James
------	--------------------	---------------	-------------

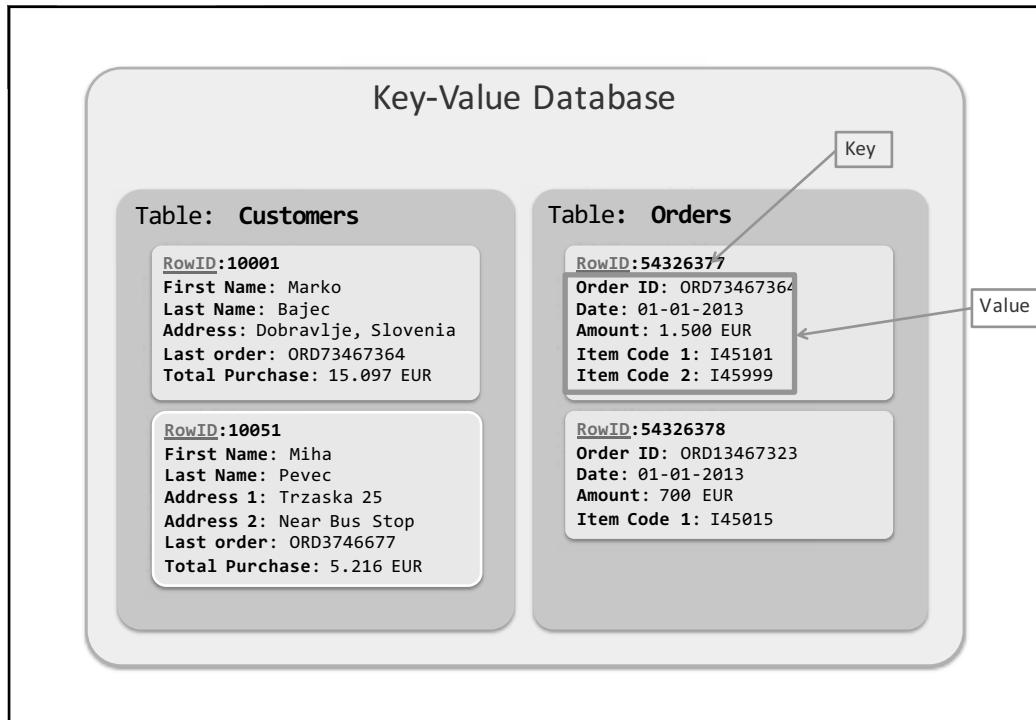
- Problem nestrukturirani in nekonsistentni podatki




- Ni sheme
- Ni praznih polj
- Uporaba SQL ni možna

Predmet: PB, Modul: Kratek pregled NoSQL paradigme, Gradivo: v.2015, ©UL FRI, Laboratorij za podatkovne tehnologije

38

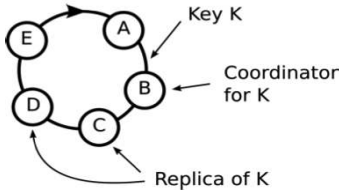


Univerza na Ljubljani
Fakulteta za računalništvo
in informatiko



NoSQL...Key-Value...Primer Dynamo

- Fragmentacija in replikacija
 - Virtualna vozlišča za heterogenost
 - Fragmentacija s pomočjo razpršitve (hash)
 - Replikacija na N predhodnih vozlišč



- Konsistentnost
 - Podprta stroga in postopna konsistentnost

Predmet: PB, Modul: Kratki pregled NoSQL paradigme, Gradivo: v.2015, ©UL FRI, Laboratorij za podatkovne tehnologije

40

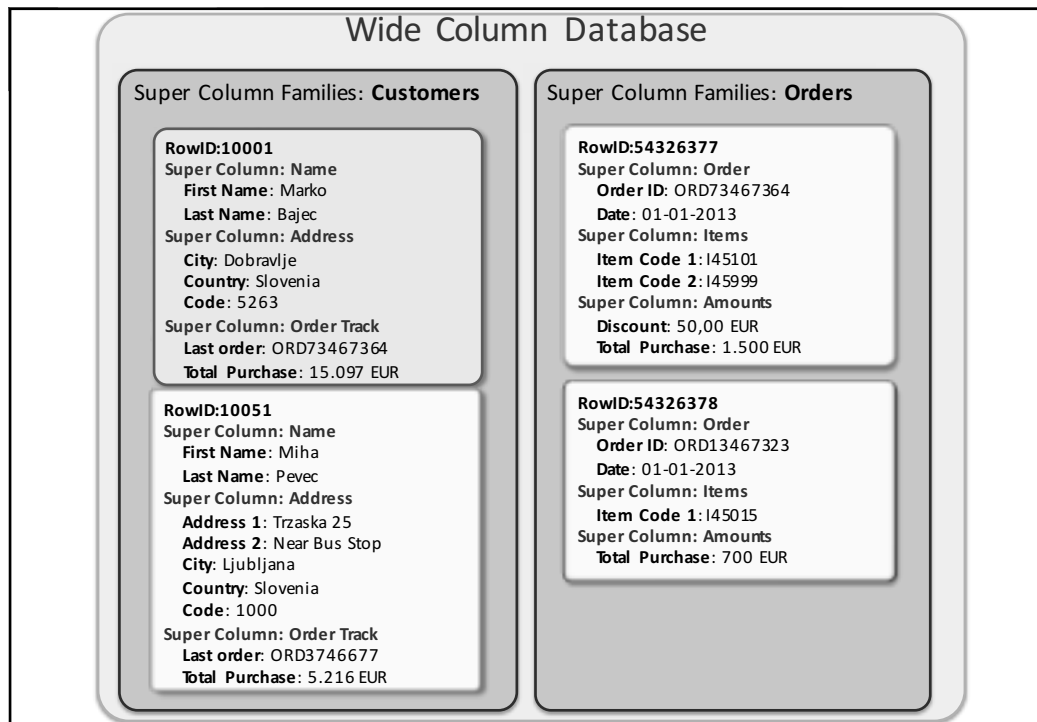
NoSQL...Key-Value...predstavniki


- Veliko drugih predstavnikov



NoSQL...Wide Column...

- Velike količine podatkov
- Stolpce združujemo v družine stolpcev → Super stolpci
- Super stolpce združujemo v družine super stolpcev → Tabele






Univerza v Ljubljani
Fakulteta za računalništvo
in informatiko

NoSQL...Wide Column...predstavniki

- Google interno
 - BigTable with MapReduce
- Odprto kodni projekti na osnovi Googlovih rešitev:
 - Hbase with Hadoop (Yahoo)
- Druge rešitve:
 - Cassandra (Facebook)



Predmet: PB, Modul: Kratki pregled NoSQL paradigme, Gradivo: v.2015, ©UL FRI, Laboratorij za podatkovne tehnologije

44

Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

NoSQL...Wide Column...primer BigTable

- Osnova za shrambo podatkov v Google App Engine.
- Podobnosti s pristopom Key-Value.
- Ključ kompoziten:
(RowKey, Family, ColumnKey, TimeStamp) → Value

family:qualifier

row-key

t1

t2

t3

Predmet: PB,

45

Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

NoSQL...Wide Column...primer BigTable

- Število stolpcev (Column) na vrstico (Row) neomejeno.
- Stolpci organizirani v družine (ColumnFamily).
- Število družin na tabelo (Table) omejeno.
- Za vsak stolpec hranimo verzije:
 - Vse verzije,
 - Zadnjih n verzij,
 - Verzije določenega časovnega obdobja
 - ...

Predmet: PB, Modul: Kratki pregled NoSQL paradigme, Gradivo: v.2015, ©UL FRI, Laboratorij za podatkovne tehnologije

46

NoSQL...Wide Column...primer BigTable

- Poizvedovanje:
 - Indeksi po ključu vrstic, ključu stolpcev, časovni znački
 - Stiki niso omogočeni
- Na voljo C++ knjižnica
 - Filtriranje po vrsticah, stolpcih, časovnih značkah
 - Iteriraje po rezultatu

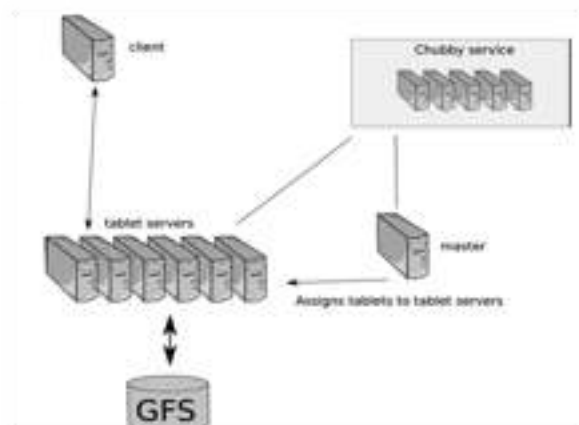
NoSQL...Wide Column...primer BigTable

- Fragmentacija:
 - Tabele razdeljene na manjše dele – Tablets.
 - Master strežnik dodeljuje Tablete posameznim strežnikom. Hrani alokacijsko tabelo.
 - Knjižnica poskrbi za predpomnjenje (caching) alokacijske tabele na odjemalcih...
- Replikacija:
 - V BigTable replikacije NI
 - Za kopije poskrbljeno na nivoju datotečnega sistema GFS Google File System.

NoSQL...Wide Column...primer BigTable

- Konsistentnost:
 - Vsak Tablet samo na enem strežniku → stroga konsistentnost...
- Arhitektura:
 - Gruča
 - Arhitektura Gospodar-Služabnik (Master-Slave)
 - Gospodar dodeljuje Tablete služabnikom.
 - Mehanizem porazdeljenega zaklepanja Google Chubby.

NoSQL...Wide Column...primer BigTable



NoSQL...dokumentne baze...

- Osnovna podatkovna enota: dokument.
- Dokument:
 - Delno strukturiran
 - Standardni formati: XML, YAML, JSON, PDF, MS Word...
 - Dokumenti le podobni - ne enaki – ni predpisane sheme

Doc1


```
{
  FirstName: "Bob",
  Address: "5 Oak St.",
  Hobby: "sailing"
}
```

Doc2

```
{
  FirstName: "Jonathan",
  Address: "15 Wanamassa Point Road",
  Children: [
    {Name: "Michael", Age: 10},
    {Name: "Jennifer", Age: 8},
    {Name: "Samantha", Age: 5},
    {Name: "Elena", Age: 2}
  ]
}
```







NoSQL...dokumentne baze...

- Organizacija dokumentov:
 - Odvisno od implementacije
 - Koncepti: Bucket, Tag, Collection....
- Ključi:
 - Vsak dokument ima svoj ključ (npr string, URI...)
 - Dokumenti so indeksirani.
- Poizvedovanje:
 - Poizvedovanje po ključu (za cel dokument)
 - Navadno tudi API ali poizvedovalni jezik za poizvedovanje po vsebini dokumentov ... Velike razlike med implementacijami!


 Univerza v Ljubljani
 Fakulteta za računalništvo
 in informatiko


NoSQL...dokumentne baze...predstavniki

- Zelo popularni dokumentni bazi:
 - MongoDB
 - CouchDB









Predmet: PB, Modul: Kratki pregled NoSQL paradigmat, Gradivo: v.2015, ©UL FRI, Laboratorij za podatkovne tehnologije

53


 Univerza v Ljubljani
 Fakulteta za računalništvo
 in informatiko

MongoDB



- Ime MongoDB izhaja iz besede "humongous"
- Odprto-kodni projekt
- Dokumentna baza
- Visoka učinkovitost (performance) in razpoložljivost (availability).
- Avtomatska razširljivost
- Iz CAP zagotovljena C in P (pri visokem A).

Predmet: PB, Modul: Kratki pregled NoSQL paradigmat, Gradivo: v.2015, ©UL FRI, Laboratorij za podatkovne tehnologije

54

MongoDB - motivacija

- Omejitve SQL SUPB
 - Nefleksibilna shema
 - Slaba razširljivost
 - Poizvedovanje zahteva stike med relacijami
- Prednosti MongoDB
 - Enostavna integracija z znanimi programskimi jeziki (npr. Java, Javascript, PHP ipd.)
 - Platformsko neodvisna
 - Ohranja ključne lastnosti relacijskih PB in vključuje dobre lastnosti key-value NoSQL PB.

MongoDB – dokumenti

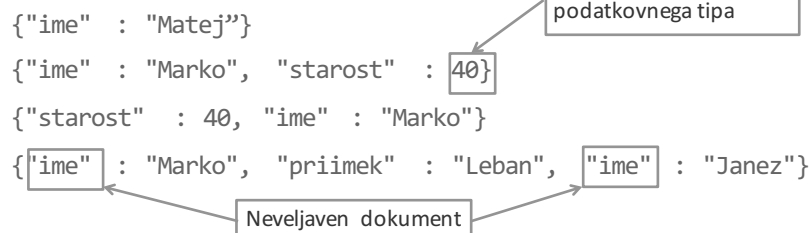
- Osnovni zapis je dokument – urejen niz vrednosti ključ-vrednost.
- Primeri dokumentov:

```

{"ime" : "Matej"}
{"ime" : "Marko", "starost" : 40}
{"starost" : 40, "ime" : "Marko"}
{"ime" : "Marko", "priimek" : "Leban", "ime" : "Janez"}
  
```

Vrednost je določenega podatkovnega tipa

Neveljaven dokument



- Pazi:
 - MongoDB je občutljiv na velike/male črke
 - Ključi se ne smejo podvajati!

MongoDB – kolekcije

- Podobne dokumente združujemo v kolekcije (collection).

```
{ "ime" : "Matej" }
{ "ime" : "Marko", "starost" : 40 }
{ "ime" : "Tina", "starost" : 10 }
{ "ime" : "Ana", "starost" : 5, "teža" : 25 }
{ "ime" : "Katja", "starost" : 22 }
```

Osebe ← Kolekcija oseb

MongoDB – shema

- MongoDB ne zahteva fiksno določene podatkovne sheme (kot npr. relacijska baza).

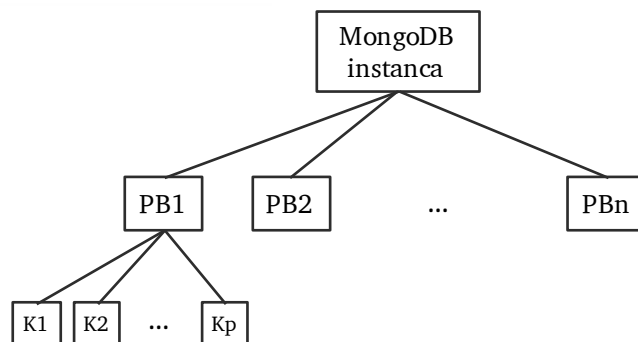
```
{ "ime" : "Matej" }
{ "ime" : "Tina", "starost" : 10 }
{ "ime" : "Ana", "starost" : 5, "teža" : 25 }
{ "ime" : "Katja", "starost" : 22 }
{ "znamka" : "citroen", "letnik" : 2009 }
```

Osebe

- MongoDB ne zahteva, da dokumente razvrščamo po kolekcijah, vendar je to priporočljivo! Uporabimo lahko tudi podkolekcije, npr. osebe.študenti

MongoDB - baza

- Baza združuje več kolekcij.
- Ena instanca MongoDB lahko vsebuje več neodvisnih baz.



Zagon MongoDB strežnika

- MongoDB strežnik lahko požene kot program ali servis v OS.
- Strežnik požene z ukazom `mongod`

\$ `mongod`

```

Markos-iMac:~ Markos$ mongod
2015-05-24T11:42:23.761+0200 I JOURNAL [initandlisten] journal dir=/data/db/journal
2015-05-24T11:42:23.761+0200 I JOURNAL [initandlisten] recover: no journal files present, no recovery needed
2015-05-24T11:42:23.772+0200 I JOURNAL [durability] Durability thread started
2015-05-24T11:42:23.772+0200 I CONTROL [initandlisten] MongoDB starting : pid=65129 port=27017 dbpath=/data/db 64-bit host=Markos-iMac.local
2015-05-24T11:42:23.772+0200 I JOURNAL [journal writer] Journal writer thread started
2015-05-24T11:42:23.773+0200 I CONTROL [initandlisten]
2015-05-24T11:42:23.773+0200 I CONTROL [initandlisten] ** WARNING: soft rlimits too low. Number of files is 256, should be at least 1000
2015-05-24T11:42:23.773+0200 I CONTROL [initandlisten] db version v3.0.2
2015-05-24T11:42:23.773+0200 I CONTROL [initandlisten] git version: nogitversion
2015-05-24T11:42:23.773+0200 I CONTROL [initandlisten] build info: Darwin Yosemite.local 14.3.0 Darwin Kernel Version 14.3.0: Mon Mar 23 11:5
9:05 PDT 2015; root:xnu-2782.20.48~5/RELEASE_ARM64_T8020 BOOST_LIB_VERSION=1_49
2015-05-24T11:42:23.773+0200 I CONTROL [initandlisten] allocator: system
2015-05-24T11:42:23.773+0200 I CONTROL [initandlisten] options: {}
2015-05-24T11:42:23.787+0200 I NETWORK [initandlisten] waiting for connections on port 27017
  
```

Zagon MongoDB odjemalca

- Odjemalec je JavaScript lupina, ki omogoča poljubne JavaScript ukaze. Odjemalca poženemo z ukazom `mongo`

\$ `mongo`

```
Markos-iMac:~ MarkoB$ mongo
MongoDB shell version: 3.0.2
connecting to: test
Server has startup warnings:
2015-05-24T11:42:23.772+0200 I CONTROL [initandlisten]
2015-05-24T11:42:23.773+0200 I CONTROL [initandlisten] ** WARNING: soft rlimits too low. Number of files is 256, should be at least 1000
>
```

```
Markos-iMac:~ MarkoB$ mongod
2015-05-24T11:42:23.761+0200 I JOURNAL [initandlisten] journal dir=/data/db/journal
2015-05-24T11:42:23.761+0200 I JOURNAL [initandlisten] recover : no journal files present, no recovery needed
2015-05-24T11:42:23.772+0200 I JOURNAL [initandlisten] Durability thread started
2015-05-24T11:42:23.772+0200 I JOURNAL [initandlisten] MongoDB starting : pid=65129 port=27017 dbpath=/data/db 64-bit host=Markos-iMac.local
2015-05-24T11:42:23.772+0200 I JOURNAL [initandlisten] Journal writer thread started
2015-05-24T11:42:23.773+0200 I CONTROL [initandlisten]
2015-05-24T11:42:23.773+0200 I CONTROL [initandlisten] ** WARNING: soft rlimits too low. Number of files is 256, should be at least 1000
2015-05-24T11:42:23.773+0200 I CONTROL [initandlisten] db version v3.0.2
2015-05-24T11:42:23.773+0200 I CONTROL [initandlisten] git version: nogitversion
2015-05-24T11:42:23.773+0200 I CONTROL [initandlisten] build info: Darwin yosemitevm.local 14.3.0 Darwin Kernel Version 14.3.0: Mon Mar 23 11:59:05 PDT 2015; root:xnu-2782.20.48~5/RELEASE_ARM_T8020 x86_64 8005T_LIB_VERSION=1.49
2015-05-24T11:42:23.773+0200 I CONTROL [initandlisten] allocator: system
2015-05-24T11:42:23.773+0200 I CONTROL [initandlisten] options: {}
2015-05-24T11:42:23.787+0200 I NETWORK [initandlisten] waiting for connections on port 27017
2015-05-24T11:44:58.170+0200 I NETWORK [initandlisten] connection accepted from 127.0.0.1:58368 #1 (1 connection now open)
```

MongoDB – dodajanje dokumenta v kolekcijo

- Postopek: kreiramo JavaScript objekt (ki je veljaven MongoDB dokument) in ga dodamo v ustrezno kolekcijo.
- Primer:
 - Kolekcija `motopark`
 - Dokument `{"tip" : "osebno", "znamka" : "Mercedes", "letnik" : 2008}`

```
> avto = {"tip" : "osebno", "znamka" : "Mercedes", "letnik" : 2008}
> {"tip" : "osebno", "znamka" : "Mercedes", "letnik" : 2008 }
> db.motopark.insert(avto)
WriteResult({"nInserted" : 1 })
> db.motopark.find()
{ "_id" : ObjectId("5561a0b05c0f4c80ff97e9a4"), "tip" : "osebno", "znamka" : "Mercedes", "letnik" : 2008 }
>
```

MongoDB – spreminjanje zapisov

- Postopek: spremenimo objekt in zamenjamo vrednost v bazi
- Primer:
 - Kolekcija motopark, dokumentu {"tip" : "osebno", "znamka" : "Mercedes", "letnik" : 2008} dodamo nov atribut "barva" z vrednostjo "modra".

```

> avto.barva = "modra"
modra
> avto
{
  "tip" : "osebno",
  "znamka" : "Mercedes",
  "letnik" : 2008,
  "barva" : "modra"
}
> db.motopark.update({znamka : "Mercedes"}, avto)
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> avto
{
  "tip" : "osebno",
  "znamka" : "Mercedes",
  "letnik" : 2008,
  "barva" : "modra"
}
> db.motopark.find()
{ "_id" : ObjectId("5561a0b05c0f4c80ff97e9a4"), "tip" : "osebno", "znamka" : "Mercedes", "letnik" : 2008, "barva" : "modra" }
>

```

MongoDB – brisanje zapisov

- Postopek: uporabimo ukaz remove
- Primer:
 - Brišemo dokument {"tip" : "osebno", "znamka" : "Mercedes", "letnik" : 2008, "barva" : "modra"}.

```

> db.motopark.remove({znamka : "Mercedes"})
WriteResult({ "nRemoved" : 1 })
>

```


MongoDB - poizvedovanje

- Na voljo ukaz `find`
- Sintaksa:
 - `db.<collection>.find`
 (`{<field1>:<value1>, <field2>:<value2>, ...}`)
- Primerjalni operatorji:
 - `<` "\$lt", `<=` "\$lte", `>` "\$gt", `>=` "\$gte"
- Primer:


```
db.motopark.find({tip:"osebno", letnik : {"$gte" : 2007}})
```

```
> db.motopark.find({tip:"osebno", letnik : {"$gte" : 2007}})
{ "_id" : ObjectId("5561a9085c0f4c80ff97e9a5"), "tip" : "osebno", "znamka" : "Mercedes", "letnik" : 2008, "barva" : "modra" }
>
```

MongoDB – drugi ukazi

```
> help
db.help()          help on db methods
db.mycoll.help()   help on collection methods
sh.help()          sharding helpers
rs.help()          replica set helpers
help admin         administrative help
help connect       connecting to a db help
help keys          key shortcuts
help misc          misc things to know
help mr            mapreduce

show dbs           show database names
show collections   show collections in current database
show users         show users in current database
show profile       show most recent system.profile entries with time >= 1ms
show logs          show the accessible logger names
show log [name]    prints out the last segment of log in memory, 'global' is default
use <db_name>      set current database
db.foo.find()      list objects in collection foo
db.foo.find( { a : 1 } ) list objects in foo where a == 1
it                result of the last line evaluated; use to further iterate
DBQuery.shellBatchSize = x set default number of items to display on shell
exit              quit the mongo shell

> █
```

MongoDB – primerjava z relacijsko bazo

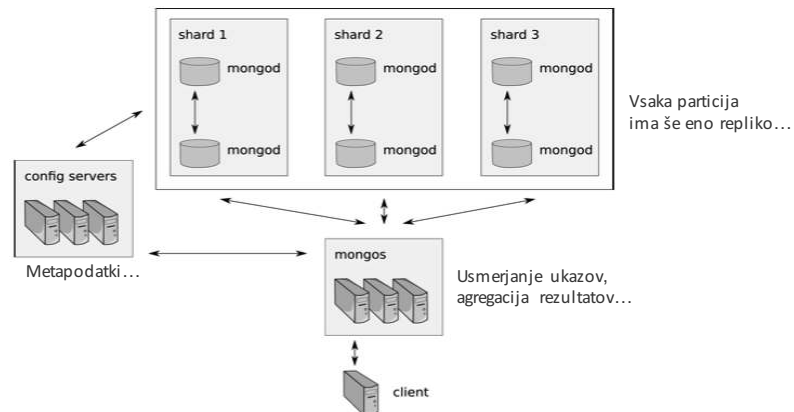
Lastnost	SQL	MongoDB
Osnovni zapis	N-terica, vrstica	Dokument
Združevanje zapisov	Relacija/tabela/pogled	Kolekcija (collection)
Ključ	PK: poljuben atribut(i)	PK: _id Field
Shema	Fiksna relacijska shema	Spremenljiva shema
Indeksiranje	Možno indeksiranje	Možno indeksiranje
Povezovanje	Stiki med relacijami	Vključena, hierarhična struktura
Porazdelitev	Particija	Shard

MongoDB – replikacija in fragmentacija

- Replikacija:
 - Različne možnosti:
 - Master Slave
 - Replica Pairs
 - Master Master Limited
 - Glavno vozlišče (master) hrani dnevnik vseh zapisovanj, ki omogoča sinhronizacijo ostalih vozlišč.
- Fragmentacija:
 - Po zbirkah ali kot določi uporabnik

MongoDB – arhitektura

Primer arhitekture MongoDB postavitev



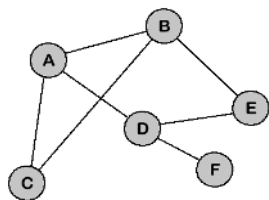
Predmet: PB, Modul: Kratki pregled NoSQL paradigme, Gradivo: v.2015, ©UL FRI, Laboratorij za podatkovne tehnologije

69

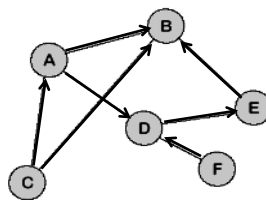
NoSQL...na osnovi grafov...

- Podatkovne baze, ki so specializirane za shranjevanje podatkov, predstavljenih v obliki grafov.

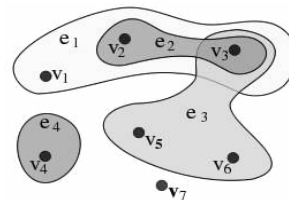
Neusmerjen graf



Usmerjen graf



Hipergraf

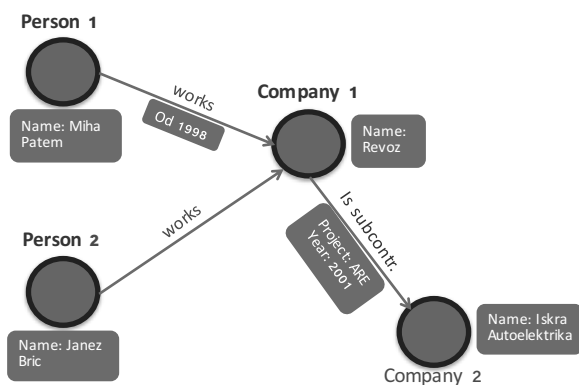
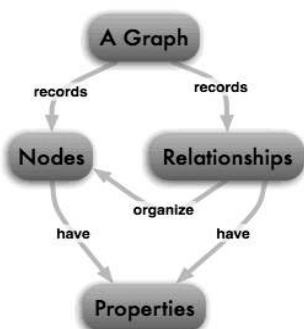


Predmet: PB, Modul: Kratki pregled NoSQL paradigme, Gradivo: v.2015, ©UL FRI, Laboratorij za podatkovne tehnologije

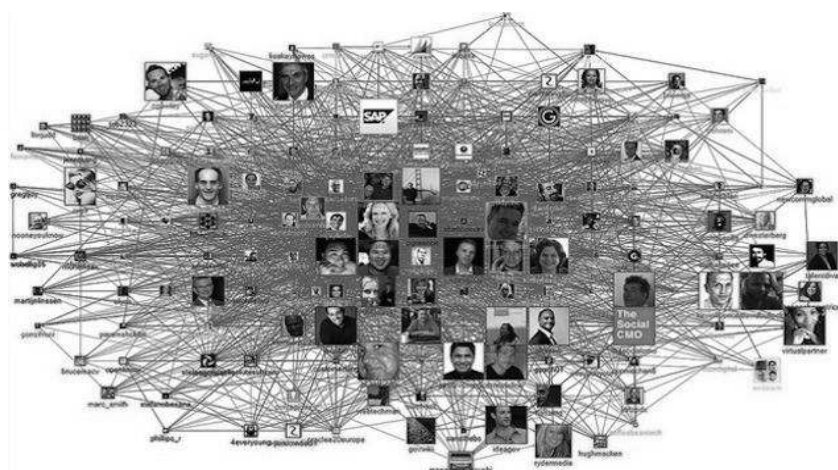
70

NoSQL...na osnovi grafov...

- Vozlišča in povezave imajo lahko lastnosti.



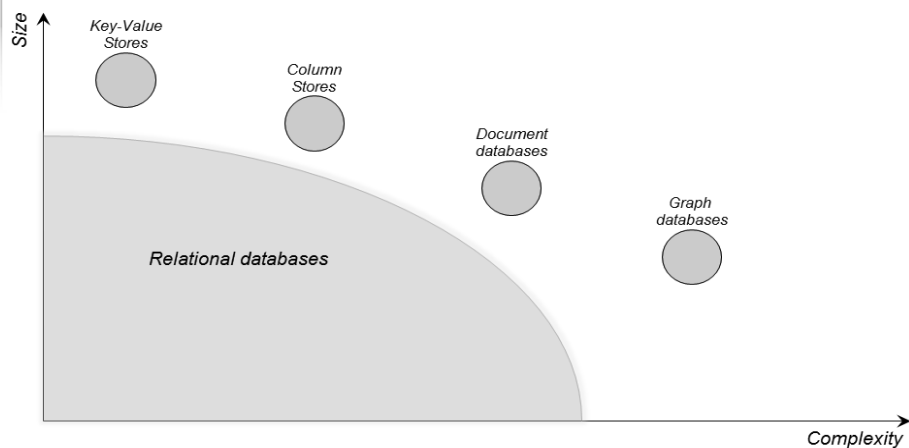
NoSQL...na osnovi grafov...



NoSQL...na osnovi grafov...

- Lastnosti:
 - Specializirana shramba za hranjenje podatkovnih struktur, s katerimi opisujemo grafe.
 - Shramba optimizirana za sprehajanje po grafu brez uporabe indeksov...;
 - Namesto globalnih poizvedb, iskanje iz izbranih vozišč navzven;
 - Fleksibilen podatkovni model – podatkovni tipi vozlišč in povezav nedefinirani;
 - Močno teoretično ozadje – teorija grafov... številni algoritmi (shortest path, Dijkstra, A*, betweenness, ...)

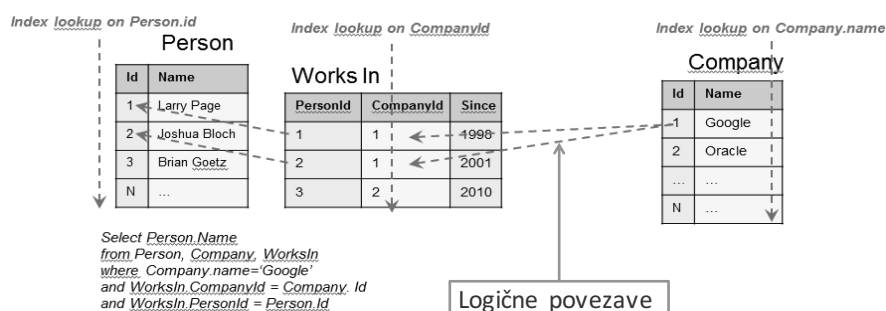
NoSQL...na osnovi grafov...umestitev



NoSQL...na osnovi grafov...primerjava...

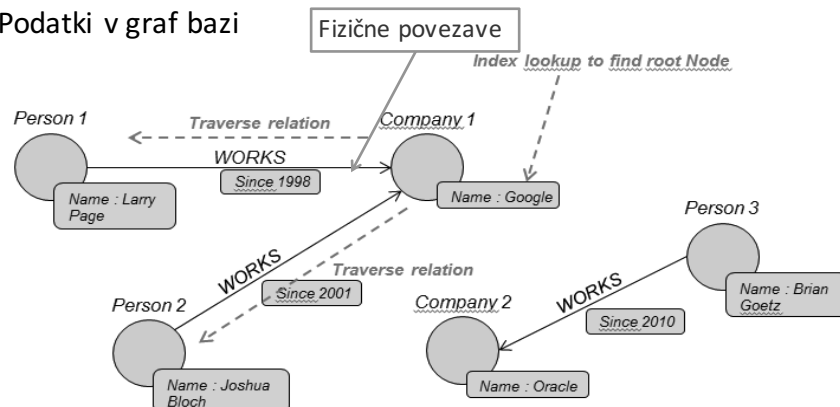
Izpiši zaposlene v podjetju Google:

Podatki v relacijski bazi



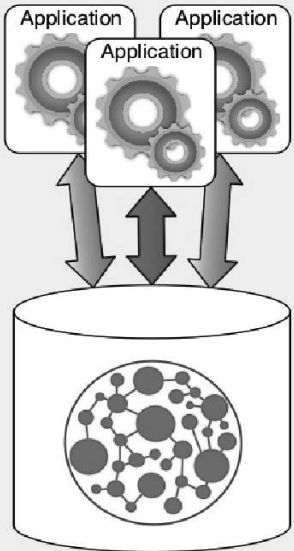
NoSQL...na osnovi grafov...primerjava


Podatki v graf bazi




DISK-BASED GRAPHS

Graph Database






Neo4j
 the graph database

Neo4j
<http://neo4j.org/>



InfiniteGraph
<http://objectivity.com>


OrientDB
 OrientDB
<http://orientdb.org>

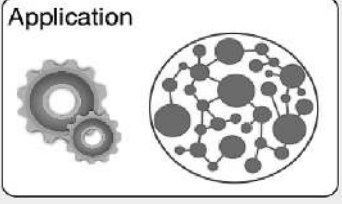

DEX

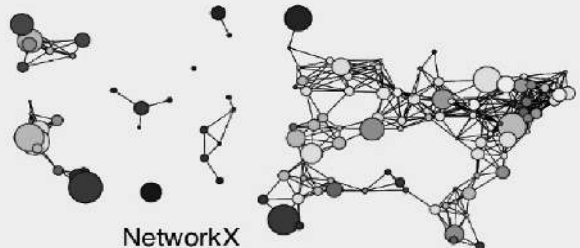
<http://www.sparsity-technologies.com/dex>

MEMORY-BASED GRAPHS

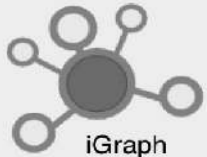
Graph Framework

Application



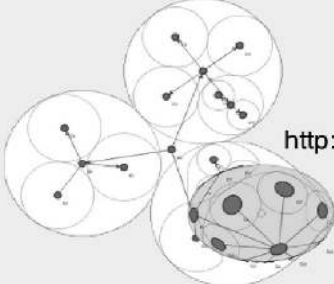


NetworkX
<http://networkx.lanl.gov/>



iGraph

<http://igraph.sourceforge.net/>



JUNG
<http://jung.sourceforge.net/>

