

NoSQL podatkovne baze

Kratek pregled



NoSQL podatkovne baze



Kaj si bomo pogledali?

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

BigData

BigData...področja

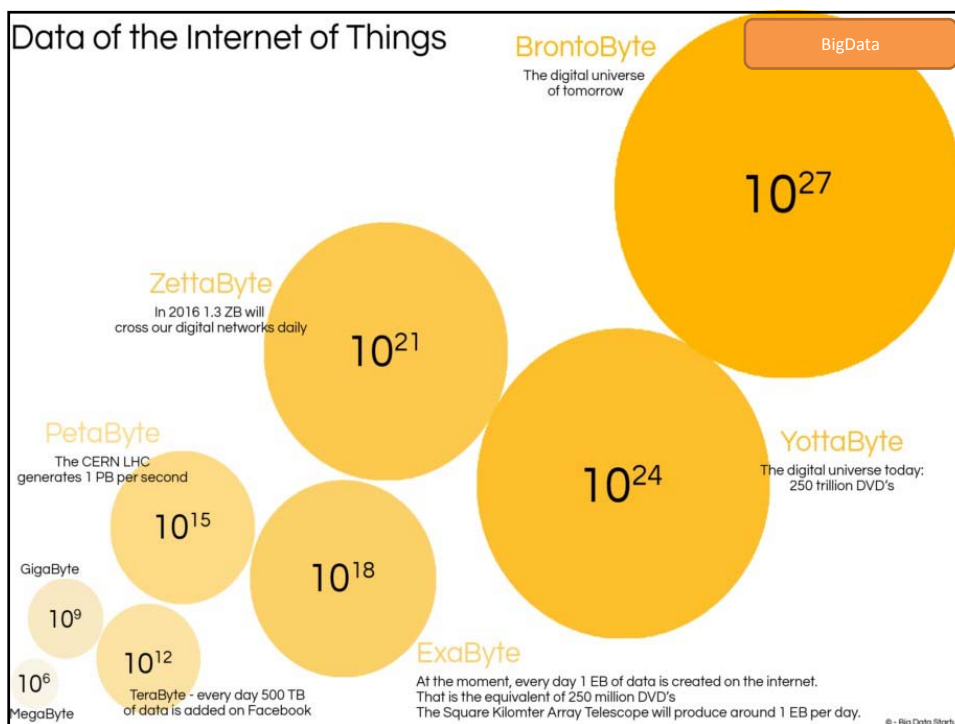
- Številni primeri, kjer imamo opravka z ogromnimi količinami podatkov...
- Senzorska omrežja,
- Družbena omrežja,
- Spletni dokumenti
- Indeksiranje spleta
- CDR
- Astronomija
- Genomika
- Vojaški nadzor
- Video arhivi
- ...



PODATKOVNE BAZE 2

- 3 -







BigData

BigData...primeri

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




- Pričakujemo 100% razpoložljivost in zanesljivost...

PODATKOVNE BAZE 2
- 5 -


BigData

BigData...razširljivost

- 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

PODATKOVNE BAZE 2
- 6 -


BigData

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



PODATKOVNE BAZE 2

Relacijske vs NoSQL

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!

PODATKOVNE BAZE 2

- 8 -


Relacijske vs NoSQL

Relacijske vs NoSQL baze...

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

PODATKOVNE BAZE 2

- 9 -




Relacijske vs NoSQL

Relacijske vs NoSQL baze

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

PODATKOVNE BAZE 2

- 10 -




Lastnosti in tehnologije porazdeljenih baz

Lastnosti in tehnologije porazdeljenih baz

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

PODATKOVNE BAZE 2

- 11 -




Lastnosti in tehnologije porazdeljenih baz

Replikacija in fragmentacija

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

PODATKOVNE BAZE 2

- 12 -



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 (Atomicity)
 - Konsistentnost (Consistency)
 - Izolacija (Isolation)
 - Trajnost (Durability)
- Relacijske baze uporabljajo 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)
 - zagotovi, da vsaka zahteva dobi odgovor o svoji uspešnosti znotraj pričakovanega časovnega intervala
- Konsistentnost (consistency) – vsa vozlišča ob istem času vidijo iste podatke
- Toleranca na nastanek particij (partition tolerance) – sistem deluje naprej kljub izgubi sporočil ali napake na delu sistema



BASE

- Model BASE kot alternativa ACID in posledica CAP:

Basically Available – sistem deluje ves čas (navzven)

Soft state – ni potrebno, da je sistem ves čas v konsistentnem stanju

Eventually consistent – po določenem času se vzpostavi konsistentnost



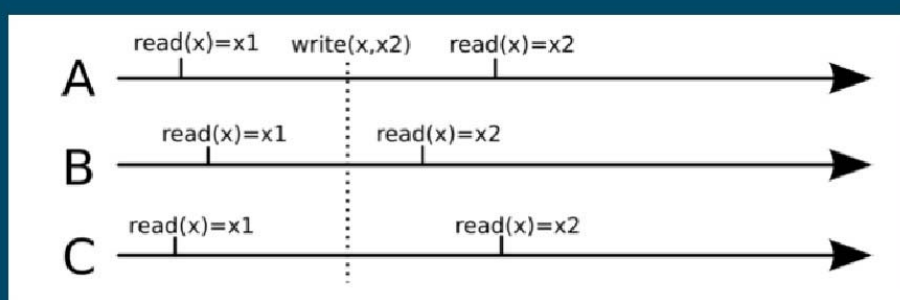
Konsistentnost...vrste

- 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



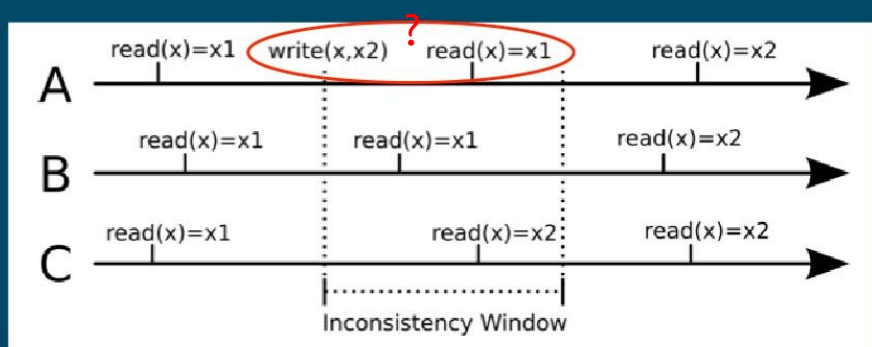
Konsistentnost...stroga

- Stroga konsistenost (Strict Consistency)



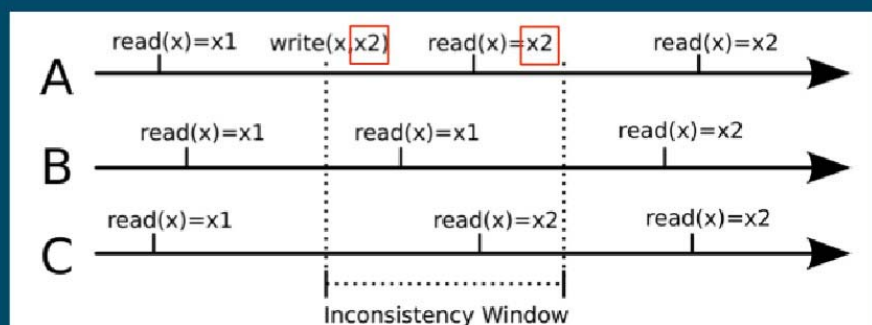
Konsistentnost...postopna...

- Postopna konsistenost (Eventual Consistency)



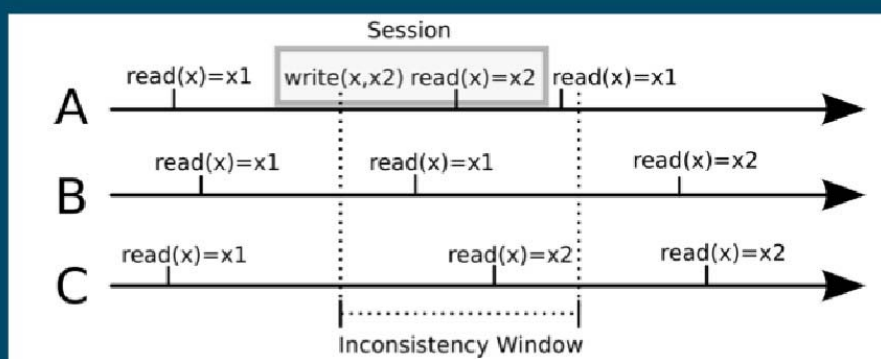
Konsistentnost...postopna

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



Konsistentnost...seje

- Konsistenost seje (Session Consistency)

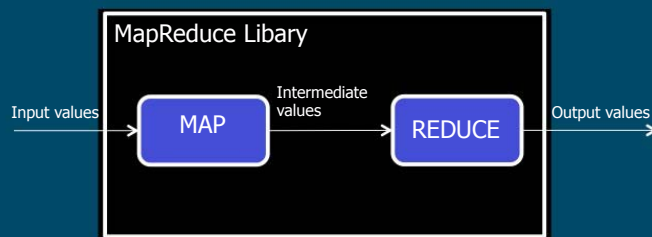


MapReduce...programski model...

- 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

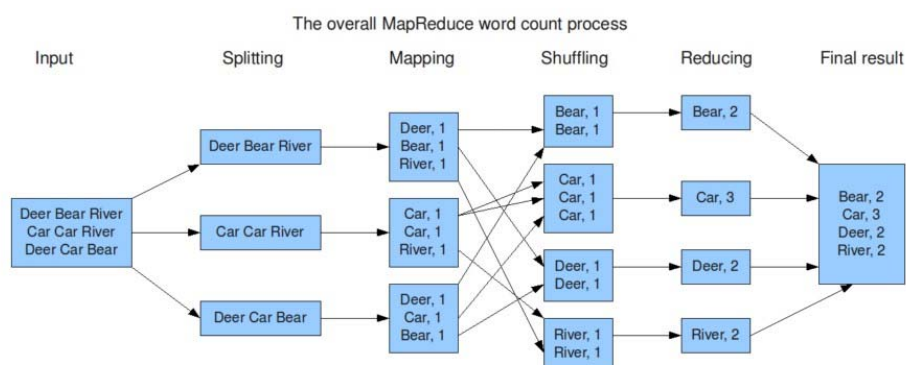


```

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

MapReduce...primer1

- Primer: štetje besed v dokumentih



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

MapReduce...primer2...

- Primer: štetje komentarjev v blogih, ki so sestavljeni iz več dokumentov (postov):

```
{
  "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": "..."
    }
  ]
}
```

- Vsak post ima svoje komentarje, nas pa zanima število komentarjev v posameznem blogu

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

PODATKOVNE BAZE 2

- 26 -



MapReduce...primer2...

- Pristop map/reduce lahko realiziramo s parom funkcij, ki deluje nad seznamom podatkov; primer je zapisan s poizvedbo LINQ, ki je integriran tudi v C#

```
from post in docs.posts
select new {
    post.blog_id,
    comments_length = comments.length
};

from agg in results
group agg by agg.key into g
select new {
    agg.blog_id,
    comments_length = g.Sum(x=>x.comments_length)
};
```

map

reduce



MapReduce...primer2...

Lastnosti in tehnologije porazdeljenih baz

1. korak: Map – za vsak dokument

```

{
  "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": "..."
    }
  ]
}

```

Step 1

Doc Id	Blog Id	Total Comments
1	3	3
2	5	4
3	6	6
4	7	1
5	3	3
6	3	5
7	2	8
8	4	3
9	5	2
10	3	3
11	5	1

```

from post in docs.posts
select new {
  post.blog_id,
  comments_length = comments.length
};

```

PODATKOVNE BAZE 2

- 27 -

MapReduce...primer2...

Lastnosti in tehnologije porazdeljenih baz

2. korak: Reduce – vhod razdelimo na particije in obdelamo vsako particijo zase (v našem primeru imamo na izhodu skupine po tri) => iz 11 vrstic na 10

Step 1

Doc Id	Blog Id	Total Comments
1	3	3
2	5	4
3	6	6
4	7	1
5	3	3
6	3	5
7	2	8
8	4	3
9	5	2
10	3	3
11	5	1

Step 2

```

from agg in results
group agg by agg.key into g
select new {
  agg.blog_id,
  comments_length = g.Sum(x=>x.comments_length)
};

```

Batch #1 Origin 1 - 3

Agg Id	Blog Id	Total Comments
1.1	3	3
1.2	5	4
1.3	6	6

Batch #2 Origin 4 - 7

Agg Id	Blog Id	Total Comments
2.1	7	1
2.2	3	8
2.3	2	8

Batch #3 Origin 8 - 10

Agg Id	Blog Id	Total Comments
3.1	4	3
3.2	5	2
3.3	3	3

Batch #4 Origin 11

Agg Id	Blog Id	Total Comments
4.1	5	1

PODATKOVNE BAZE 2

Lastnosti in tehnologije porazdeljenih baz

MapReduce...primer2...

3. korak: Reduce – nadaljujemo
=> 10 vrstic na 7

Step 2

Batch #1	Origin 1 - 3		
Agg Id	Blog Id	Total Comments	
1.1	3	3	
1.2	5	4	
1.3	6	6	

Batch #2

Origin 4 - 7		
Agg Id	Blog Id	Total Comments
2.1	7	1
2.2	3	8
2.3	2	8

Batch #3

Origin 8 - 10		
Agg Id	Blog Id	Total Comments
3.1	4	3
3.2	5	2
3.3	3	3

Batch #4

Origin 11		
Agg Id	Blog Id	Total Comments
4.1	5	1

Step 3

Batch #1	Origin 1 - 3		
Agg Id	Blog Id	Total Comments	
1.1	3	14	
1.2	5	6	
1.3	6	6	
1.4	7	1	
1.5	2	8	
1.6	4	3	

Batch #2

Origin 4		
Agg Id	Blog Id	Total Comments
2.1	5	1

```

from agg in results
group agg by agg.key into g
select new {
    agg.blog_id,
    comments_length = g.Sum(x=>x.comments_length)
};

```

Izhod mora imeti enak format kot vhod, da lahko reduce ponovimo!

PODATKOVNE BAZE 2

Lastnosti in tehnologije porazdeljenih baz

MapReduce...primer2...

4. korak: Reduce – nadaljujemo
=> 7 vrstic na 6
Konec, ker so vsi Blog ID unikatni.

Štép 3

Batch #1	Origin 1 - 3		
Agg Id	Blog Id	Total Comments	
1.1	3	14	
1.2	5	6	
1.3	6	6	
1.4	7	1	
1.5	2	8	
1.6	4	3	

Batch #2

Origin 4		
Agg Id	Blog Id	Total Comments
2.1	5	1

Step 4

Batch #1	Origin 1 - 2		
Agg Id	Blog Id	Total Comments	
1.1	3	14	
1.2	5	7	
1.3	6	6	
1.4	7	1	
1.5	2	8	
1.6	4	3	

```

from agg in results
group agg by agg.key into g
select new {
    agg.blog_id,
    comments_length = g.Sum(x=>x.comments_length)
};

```

Kaj se zgodi, če dodam nov komentar v post z id=10?

PODATKOVNE BAZE 2

MapReduce...primer2

Kaj se zgodi, če dodam nov komentar v post z id=10?

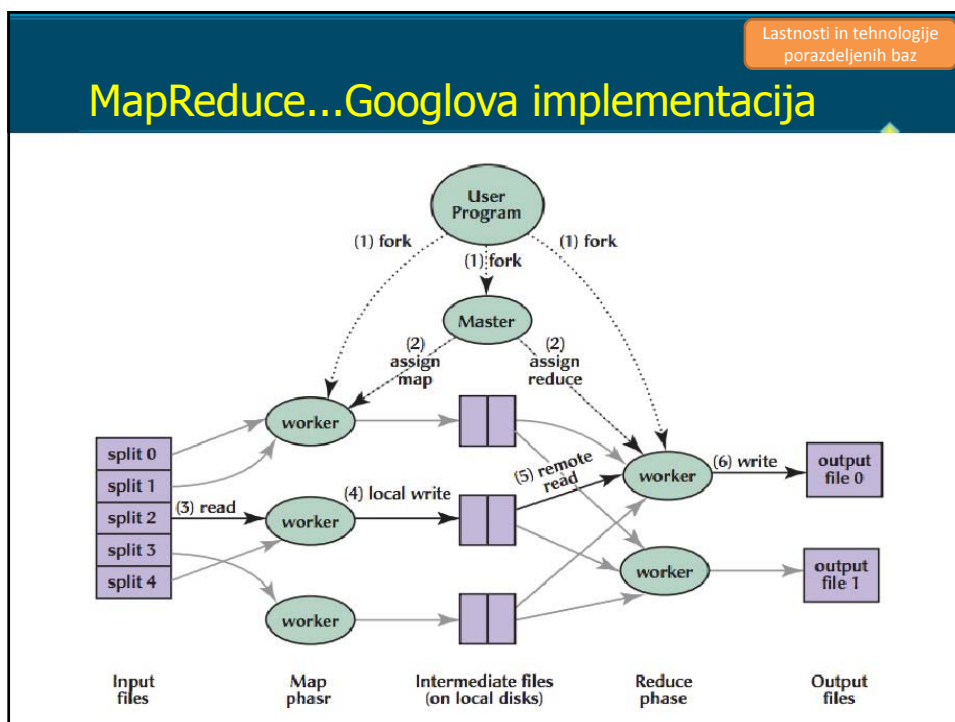
1. Rezultat poizvedbe ni več pravilen.
2. VENDOR: ni nam potrebno ponovno obdelati vseh podatkov!
3. Zadošča:
 - a. Map post z id=10
 - b. Reduce Korak 2 – samo Batch 3
 - c. Reduce Korak 3 – samo Batch 1
 - d. Reduce Korak 4



MapReduce...implementacije

- Številne različne implementacije
 - Večprocesorski strežniki z NUMA pomnilnikom
 - 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
 - Repliacija
 - Poseben sistem za razvrščanje opravil





Lastnosti in tehnologije porazdeljenih baz

MapReduce...Hadoop

- Apache, odprto-kodni projekt
- HDFS - Hadoop Distributed File System
 - Porazdeljen datotečni sistem
 - Sistemsko nezahteven (namestitev na cenenih računalnikih)
 - Visoko odporen na odpovedi
 - Visoka propustnost
- Hadoop YARN
 - Orodje za dodeljevanje opravil in upravljanje s cluster-i
- Hadoop MapReduce
 - Sistem za paralelno procesiranje velikih količin podatkov temelječe na YARN





PODATKOVNE BAZE 2

- 33 -

NoSQL podatkovne baze

PODATKOVNE BAZE 2

NoSQL podatkovne baze

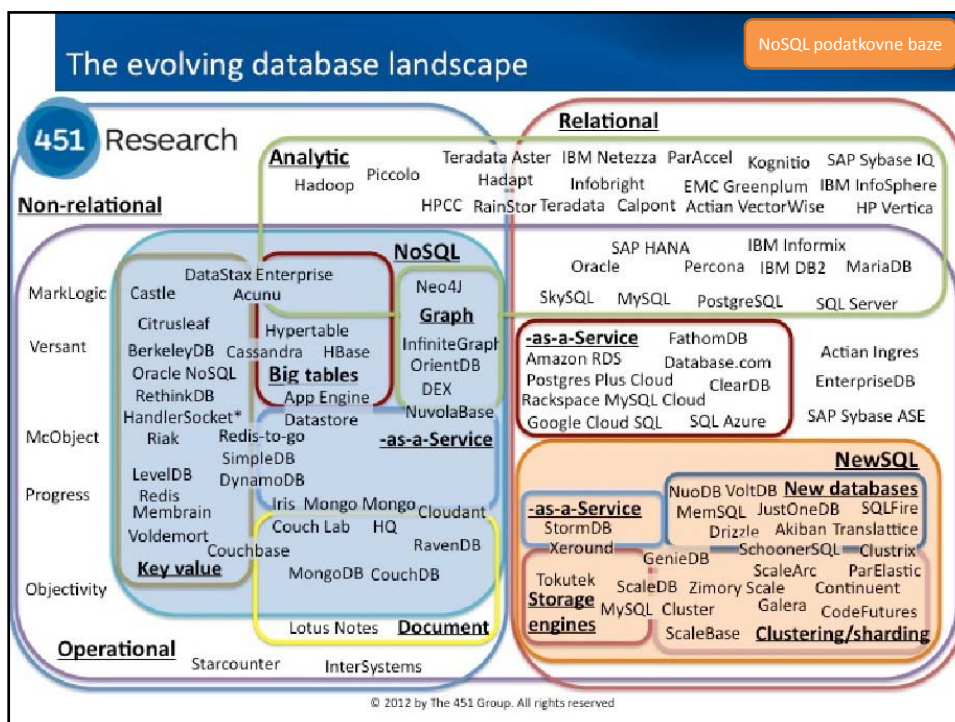
NoSQL...uvrstitev...

- Številni načini uvrstitve NoSQL
- Trenutni trendi:
 - NoSQL PB za razširljivost na porazdeljenih arhitekturah, podatki brez sheme...
 - NewSQL PB za porazdeljene arhitekture z zagotavljanjem ACID in podporo SQL...
 - 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>

PODATKOVNE BAZE 2

- 35 -



NoSQL podatkovne baze

NoSQL...vrste

- **Key Value Store**
 - Azure Table Storage, MEMBASE, GenieDB, Tokyo Cabinet / Tyrant, MemcacheDB, 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/>

PODATKOVNE BAZE 2

- 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
1		

Key:	Email:	Location:	Age:
2	j.b@gmail.com	London	37

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

- Problem nestrukturirani in nekonsistentni podatki



Key-Value Database

Table: Customers

RowID:10001
 First Name: Janko
 Last Name: Novak
 Address: Dobravlje, Slovenia
 Last order: ORD73467364
 Total Purchase: 15.097 EUR

RowID:10051
 First Name: Miha
 Last Name: Pevec
 Address 1: Trzaska 25
 Address 2: Near Bus Stop
 Last order: ORD3746677
 Total Purchase: 5.216 EUR

Table: Orders

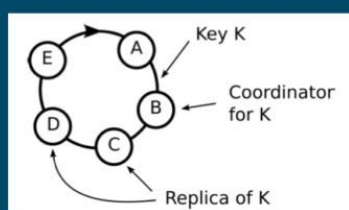
RowID:54326377 ← Key
 Order ID: ORD73467364
 Date: 01-01-2013
 Amount: 1.500 EUR
 Item Code 1: I45101
 Item Code 2: I45999 ← Value

RowID:54326378
 Order ID: ORD13467323
 Date: 01-01-2013
 Amount: 700 EUR
 Item Code 1: I45015

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

- Fragmentacija in replikacija

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



- Konsistentnost

- Podprta stroga in postopna konsistentnost

NoSQL...Key-Value...predstavniki

- Veliko drugih predstavnikov



NoSQL podatkovne baze

NoSQL...Wide Column...

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

- 42 -

PODATKOVNE BAZE 2

NoSQL podatkovne baze

Wide Column Database

Super Column Families: Customers

RowID:10001
Super Column: Name
First Name: Janko
Last Name: Novak
Super Column: Address
City: Dobravlje
Country: Slovenia
Code: 5263
Super Column: Order Track
Last order: ORD73467364
Total Purchase: 15.097 EUR

RowID:10051
Super Column: Name
First Name: Miha
Last Name: Pevec
Super Column: Address
Address 1: Trzaska 25
Address 2: Near Bus Stop
City: Ljubljana
Country: Slovenia
Code: 1000
Super Column: Order Track
Last order: ORD3746677
Total Purchase: 5.216 EUR

Super Column Families: Orders

RowID:54326377
Super Column: Order
Order ID: ORD73467364
Date: 01-01-2013
Super Column: Items
Item Code 1: I45101
Item Code 2: I45999
Super Column: Amounts
Discount: 50,00 EUR
Total Purchase: 1.500 EUR

RowID:54326378
Super Column: Order
Order ID: ORD13467323
Date: 01-01-2013
Super Column: Items
Item Code 1: I45015
Super Column: Amounts
Total Purchase: 700 EUR

PODATKOVNE BAZE 2

NoSQL...Wide Column...predstavniki

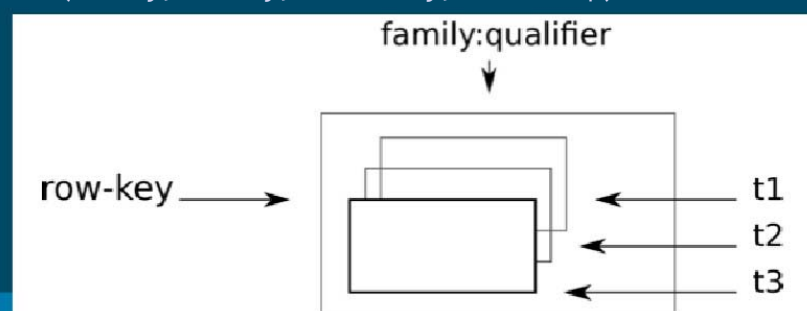
- Google interno
 - BigTable with MapReduce
- Odprto kodni projekti na osnovi Googlovih rešitev:
 - Hbase with Hadoop (Yahoo)
- Druge rešitve:
 - Cassandra (Facebook)



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



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
 - ...

NoSQL...Wide Column...primer BigTable

- Poizvedovanje:
 - Indeksi po ključu vrstic, ključu stolpcev, časovni znački
 - Stiki niso omogočeni
 - 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 srežnikom. 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:

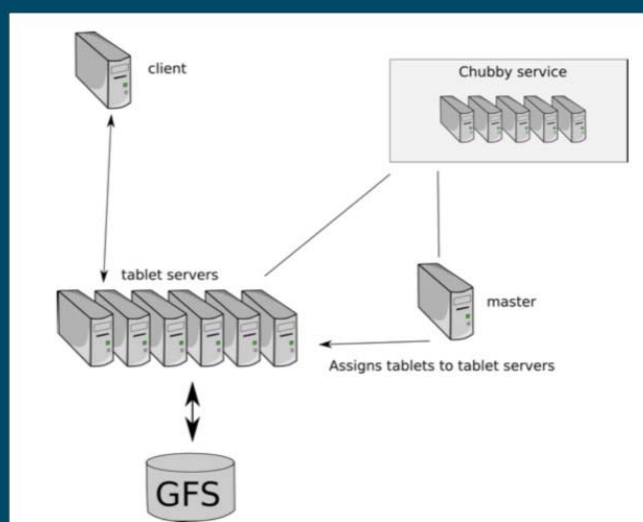
- Vsak Tablet samo na enem strežniku → stroga konsistentnost...

- 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... binarni formati 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
- 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!

NoSQL...dokumentne baze...predstavniki


- Zelo popularni dokumentni bazi:
 - MongoDB
 - CouchDB



NoSQL podatkovne baze

NoSQL...dokumentne baze...MongoDB

- **Osnovne lastnosti**
 - Ne uporablja sheme
 - Odprtokodni projekt
 - Koda v C++ ... MongoDB hiter in razširljiv
 - Omogoča tudi hranjenje velikih binarnih datotek (video, slike...)
- **Dokumenti**
 - BSON (Binary JSON) objekti
 - Vsak dokument ima ključ
 - Podporto gnezdenje (dokument v dokumentu)
 - Indeksiranje tudi po vgnezenih dokumentih




PODATKOVNE BAZE 2
- 54 -

NoSQL podatkovne baze

NoSQL...dokumentne baze...MongoDB

- **Organizacija dokumentov:**
 - Dokumenti organizirani v zbirke (Collections)
 - Zbirke lahko vsebujejo poljubne dokumente.
 - Poizvedbe se vedno nanašajo na eno zbirko.
- **Poizvedovanje:**
 - JSON sintaksa
 - Poizvedovanje po zbirki vključno z gnezdenimi dokumenti
 - Podpira agregacije z uporabo posebne različice MapReduce implementacije.



PODATKOVNE BAZE 2
- 55 -

NoSQL...dokumentne baze...MongoDB

- Replikacija:

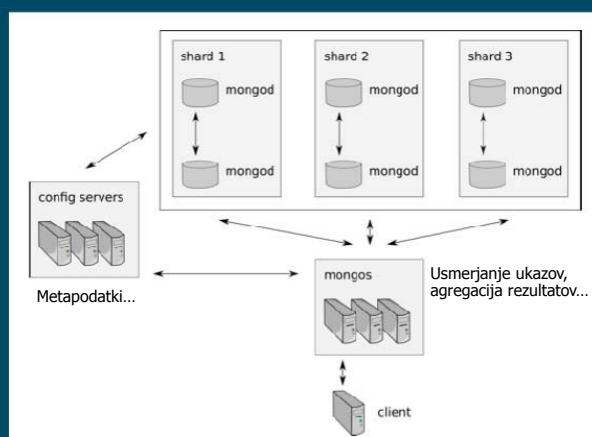
- Različne možnosti:
 - Master Slave
 - Replica Pairs
 - Master Master Limited
- Glavno vozlišče (master) hrani dnevnik vseh zapisovanj, ki omogoča sinhronizacijo ostalih vozlišč.

- Fragmentacija:

- Po zbirkah ali kot določi uporabnik

NoSQL...dokumentne baze...MongoDB

- Arhitektura

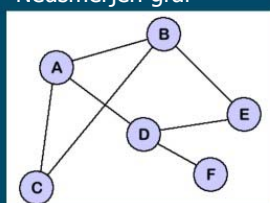


NoSQL...na osnovi grafov...

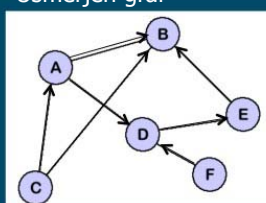
- Podatkovne baze, ki so specializirane za shranjevanje podatkov, predstavljenih v obliki grafov.



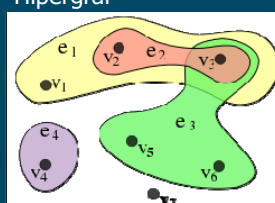
Neusmerjen graf



Usmerjen graf

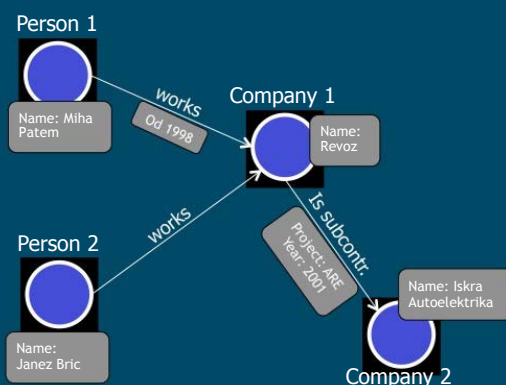
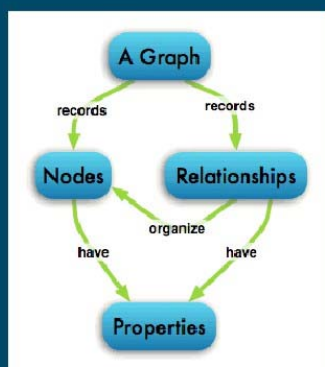


Hipergraf

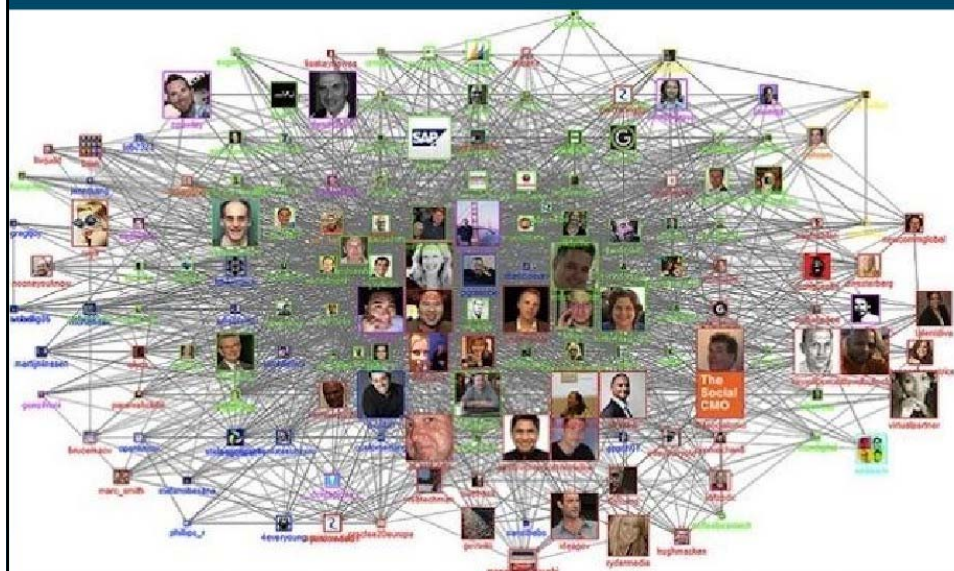


NoSQL...na osnovi grafov...

- Vozlišča in povezave imajo lahko lastnosti.



NoSQL...na osnovi grafov...



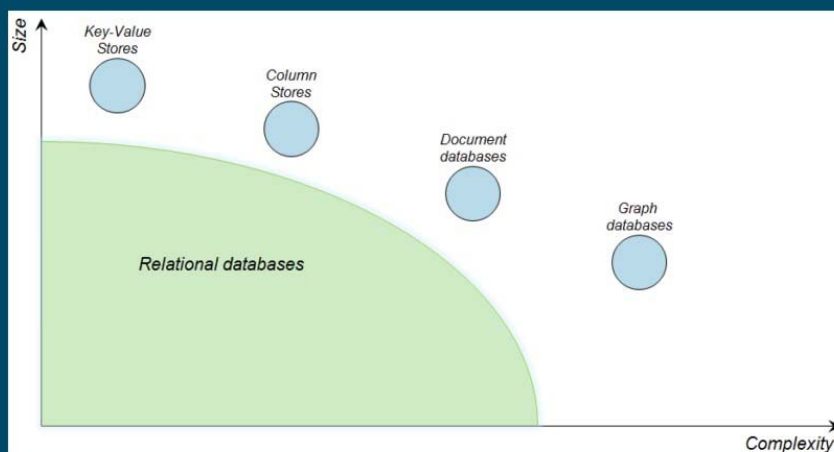
NoSQL...na osnovi grafov...

- Lastnosti:

- Shramba specializirana 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 vozišč in povezav nedefinirani;
- Močno teoretično ozadje - teorija grafov... številni algoritmi (shortest path, Dijkstra, A*, betweenness, ...)



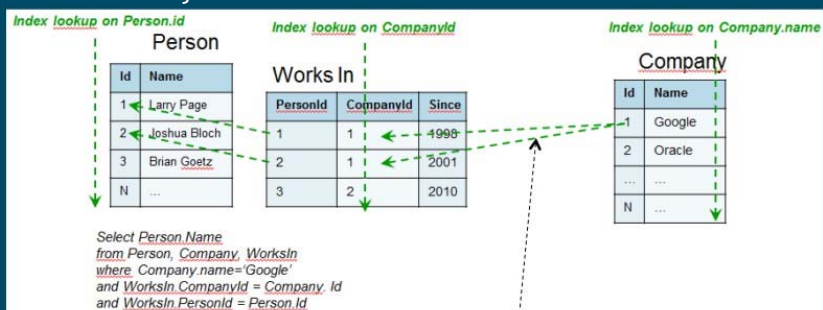
NoSQL...na osnovi grafov...umestitev



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

Izpiši zaposlene v podjetju Google:

Podatki v relacijski bazi



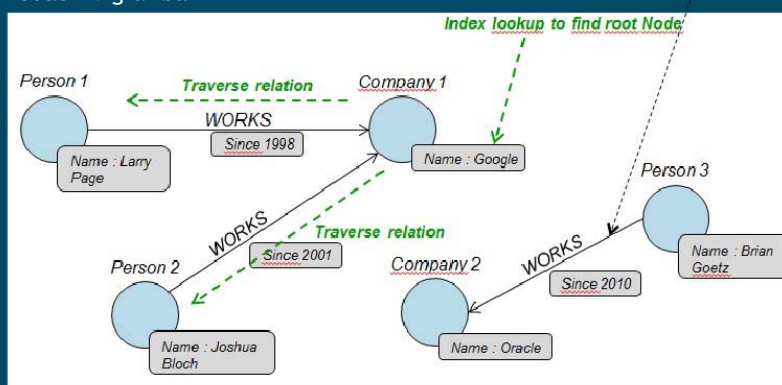
Logične povezave

NoSQL...na osnovi grafov...primerjava

Izpiši zaposlene v podjetju Google:

Podatki v graf bazi

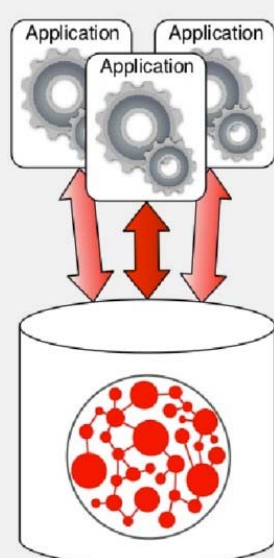
Fizične povezave



- 64 -

DISK-BASED GRAPHS

Graph Database



Neo4j
the graph database

Neo4j
<http://neo4j.org/>



InfiniteGraph
<http://objectivity.com>

OrientDB
OrientDB
<http://orientdb.org>

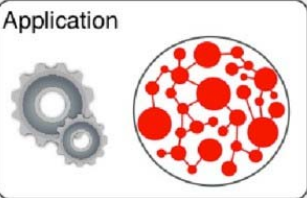
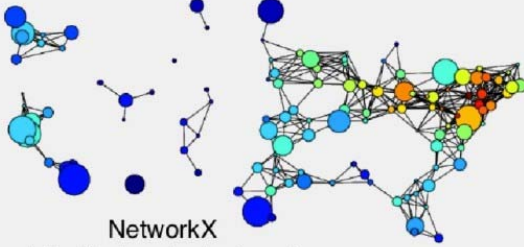
***dex**
DEX

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

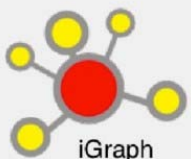
MEMORY-BASED GRAPHS

Graph Framework

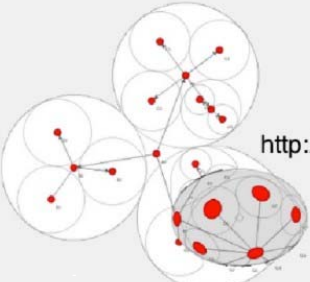
Application

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



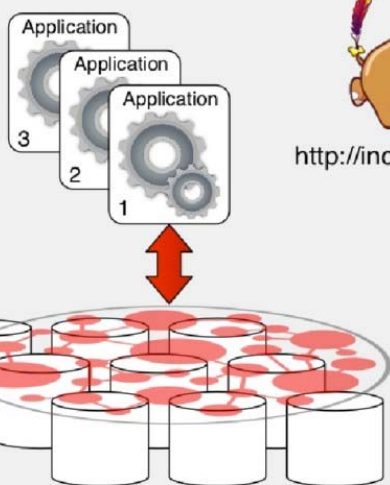

iGraph
<http://igraph.sourceforge.net/>




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

CLUSTER-BASED GRAPHS


Bulk Synchronous Parallel Processing

Hama
<http://incubator.apache.org/hama/>



Giraph
<http://incubator.apache.org/giraph/>



GOLDENORB
GoldenOrb
<http://goldenorbos.org/>