



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Esempio di NoSQL: MongoDB

Basi di Dati

Corso di Laurea in Informatica per il Management

Alma Mater Studiorum - Università di Bologna

Prof. Marco Di Felice

Dipartimento di Informatica – Scienza e Ingegneria

marco.difelice3@unibo.it

Introduzione ai DBMS NoSQL

NoSQL: Movimento che promuove l'adozione di **DMBS non basati sul modello relazionale**.

- Il termine NOSQL appare per la prima volta in una pubblicazione di Carlo Strozzi nel 1998.
- Oggi, il termine NOSQL viene usato per lo più nell'accezione **NoT Only SQL**.

"Next generation databases mostly addressing some of the points: being non-relational, distributed, open source and horizontally scalable" (definizione da <http://nosql-database.org/>)

Introduzione ai DBMS NoSQL

Il termine NoSQL identifica **una varietà di DBMS non relazionali**, basati su **modelli logici differenti**:

- Database **chiave/valore**



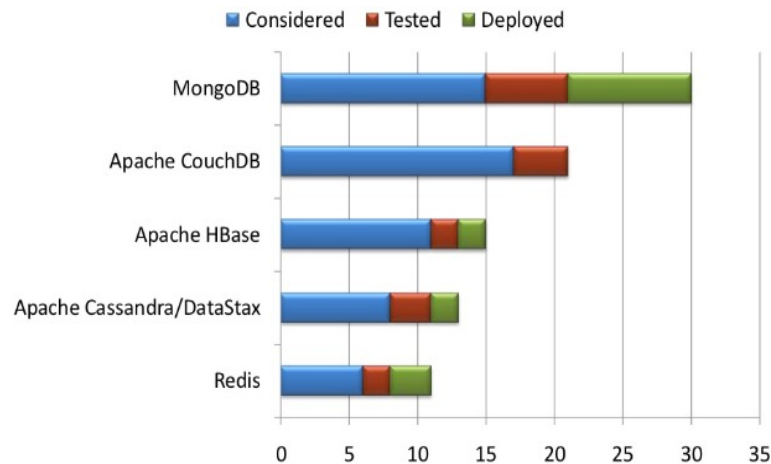
- Database **document-oriented**

- Database **column-oriented**

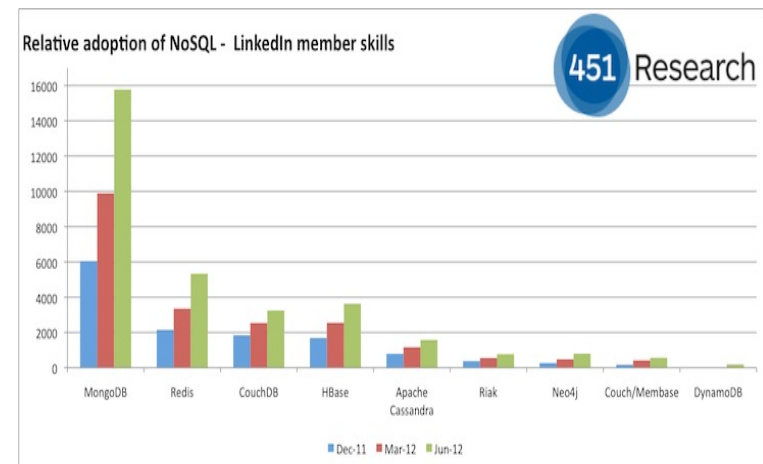
- Database **graph-oriented**

Esempio di DB NoSQL: MongoDB

- **MongoDB** (<https://www.mongodb.org>)
- DBMS non relazionale, open source dal 2009
- Utilizzi: <https://www.mongodb.com/who-uses-mongodb>



Source: <http://www.datasciencecentral.com/>



Source: <https://blogs.the451group.com/>

Esempio di DB NoSQL: MongoDB

- **MongoDB** (<https://www.mongodb.org>)
- Database organizzato in **collezioni**; le collezioni contengono liste di **documenti**. Ogni documento è un insieme di **campi**.

MongoDB	Modello Relazionale
<i>Collezione</i>	Tabella
<i>Documento</i>	Riga
<i>Campo</i>	Colonna di una riga

Esempio di DB NoSQL: MongoDB

- **MongoDB** (<https://www.mongodb.org>)
- Inserimento comandi **JavaScript** tramite shell o driver immersi in linguaggio ospite (Java).
- Utilizzo linguaggio **JSON** come input/output delle query di aggiornamento o selezione.
- Utilizzo linguaggio **BSON** (codifica binaria di JSON) per rappresentare i documenti internamente.

Esempio di DB NoSQL: MongoDB

JSON

- **Formato** per lo scambio di dati tra applicazioni.
- Documenti JSON facilmente interpretabili da macchine; molti parser JSON sono disponibili (*machine understandability*).
- I dati di un documento sono racchiusi tra {}.
- I dati assumono la forma: **chiave : valore**

```
{"nome": "mario" }
```

```
{"nome": "mario", "cognome": "rossi"}
```

Esempio di DB NoSQL: MongoDB

- Valore → **Numero**, intero o reale
`{"nome": "mario", "eta": 15, "punti": 13.45}`
- Valore → **Stringa**, tra apici
`{"nome": "mario", "cognome": "rossi"}`
- Valore → **Booleano**, true o false
`{"nome": "mario", "impiegato": true}`
- Valore → **Array**, tra parentesi quadre
`{"nome": "mario", "cap": ["134", "042"]}`
- Valore → **Oggetto**, tra parentesi graffe
`{"nome": "mario", "indirizzo": {"citta": "bologna", "via": "po", "numero": 3} }`

Esempio di DB NoSQL: MongoDB

```
{ "nome": "Mario",  
  "cognome": "Rossi",  
  "eta": 45,  
  "impiegato": false,  
  "salario": 1205.50,  
  "telefono": ["0243434", "064334343"],  
  "ufficio": [  
    {"nome": "A", "via": "Zamboni", "numero": 7},  
    {"nome": "B", "via": "Irnerio", "numero": 49}]  
}
```

Esempio di
DOCUMENTO
JSON

Esempio di DB NoSQL: MongoDB

- Avvio del **server** e della **shell**

```
mongod    //demone in ascolto sulla porta
mongo     //shell client
```

- **Utilizzo/Creazione** di un DB

```
use provaDB
```

- Creazione di una **collezione** (vuota)

```
db.createCollection("circoli");
```

Esempio di DB NoSQL: MongoDB

- Comandi della shell di **MongoDB**

Comando	Azione
show dbs	Mostra DB disponibili
show collections	Mostra le collezioni del db
show users	Mostra gli utenti del sistema
show rules	Mostra il sistema di accessi
show logs	Mostra i log disponibili

Esempio di DB NoSQL: MongoDB

- **Documento** in MongoDB → oggetto **JSON**!

```
{campo1: valore1, campo2: valore2,  
  campo3:[valore1, ..., valoreN]}
```

```
{"name": "Marco", "cognome": "Rossi", "eta": 22}
```

```
{"name": "Marco", "cognome": "Rossi",  
  "eta": 22, "domicilio":["Roma", "Bologna"]}
```

Esempio di DB NoSQL: MongoDB

- Nella stessa collezione, è possibile inserire documenti **eterogenei**, ossia con **strutture campo/valore** differenti.

COLLEZIONE ANAGRAFICA

DOC1	Marco		22	
DOC2	Massimo	Rossi		
DOC3	Maria	Bianchi	24	1/5/1990

- Non è possibile rappresentare casistiche simili nel modello relazionale, a meno di usare valori NULL!

Esempio di DB NoSQL: MongoDB

Inserimento di un documento in una collezione

`db.NOMECOLLEZIONE.insert(DOCUMENTO)`

```
db.anagrafica.insert({"name": "Marco", "cognome":  
"Rossi", "eta": 22})
```

```
db.anagrafica.insert({"cognome": "Rossi", "eta": 22,  
"domicilio":["Roma", "Bologna"]})
```

```
db.anagrafica.insert({"name": "Maria", "eta": 25})
```

Esempio di DB NoSQL: MongoDB

- Ogni documento contiene un campo **_id**, che corrisponde alla **chiave primaria** della collezione.
- Il campo **_id** può essere definito **esplicitamente**, o viene aggiunto in maniera implicita da MongoDB.

```
db.anagrafica.insert({"_id": 1, "name": "Marco",  
"cognome": "Rossi", "eta": 22})
```

ID ESPLICITO

```
db.anagrafica.insert({"name": "Marco", "cognome":  
"Rossi", "ruolo": "Manager"})
```

ID IMPLICITO

Esempio di DB NoSQL: MongoDB

Rimozione di un documento da una collezione

`db.NOMECOLLEZIONE.remove({})` → Svuota la collezione, eliminando tutti gli elementi.

`db.anagrafica.remove({})` →
DELETE FROM anagrafica

`db.NOMECOLLEZIONE.remove(SELETTORE)` → Eliminando dalla collezione tutti i documenti che fanno matching con il selettore

Esempio di DB NoSQL: MongoDB

- **SELETTORE** → Documento JSON

```
{campo1: valore1, campo2:valore2, ...}
```

```
{"name":"Marco", "cognome":"Rossi"}
```

```
{$or[{campo1: valore1},{campo2:valore2}]}
```

```
{$or[{"name":"Marco"}, {"cognome":"Rossi"}]}
```

```
$gt, $lt, $gte, $lte, $ne, $in
```

```
{campo1: valore1, campo2:{$OP:VALORE/I}}
```

```
{"name":"Marco", "eta":{$gt:30}}
```

Esempio di DB NoSQL: MongoDB

Aggiornamento di un documento di una collezione

```
db.NOMECOLLEZIONE.update(SELETTORE, CAMPI)
```

```
db.NOMECOLLEZIONE.update(SELETTORE, {$SET: CAMPI})
```

```
db.NOMECOLLEZIONE.update(SELETTORE, {$PUSH: CAMPI})
```

```
db.anagrafica.update({"name": "Mario"}, {"eta": 45})
```

→ Sostituisce il documento relativo all'impiegato Mario, con la coppia nome/età!

Esempio di DB NoSQL: MongoDB

Aggiornamento di un documento di una collezione

```
db.NOMECOLLEZIONE.update(SELETTORE,CAMPI)
```

```
db.NOMECOLLEZIONE.update(SELETTORE,{ $SET: CAMPI})
```

```
db.NOMECOLLEZIONE.update(SELETTORE,{ $PUSH:CAMPI})
```

```
db.anagrafica.update( ({"name": "Mario"} , { $set: {"eta":45}}) ) → Nel documento relativo all'impiegato Mario, aggiorna il campo età ponendolo pari a 45.
```

Esempio di DB NoSQL: MongoDB

Aggiornamento di un documento di una collezione

```
db.NOMECOLLEZIONE.update(SELETTORE,CAMPI)
```

```
db.NOMECOLLEZIONE.update(SELETTORE,{$SET: CAMPI})
```

```
db.NOMECOLLEZIONE.update(SELETTORE,{$PUSH:CAMPI})
```

```
db.anagrafica.update(({"name": "Mario"},{$set:{"eta":45},{multi:true}}) → in tutti i documenti relativi all'impiegato Mario, si aggiorna il campo età ponendolo pari a 45.
```

Esempio di DB NoSQL: MongoDB

Aggiornamento di un documento di una collezione

```
db.NOMECOLLEZIONE.update(SELETTORE, CAMPI)
```

```
db.NOMECOLLEZIONE.update(SELETTORE, {$SET: CAMPI})
```

```
db.NOMECOLLEZIONE.update(SELETTORE, {$PUSH: CAMPI})
```

`db.anagrafica.update({"name": "Mario"}, {$push: {"eta": 45}})` → Nel documento relativo all'impiegato Mario, si aggiunge un nuovo campo età (array), settandolo pari a 45.

Esempio di DB NoSQL: MongoDB

Ricerca di un documento all'interno di una collezione

`db.NOMECOLLEZIONE.find()` → restituisce tutti i documenti presenti nella collezione.

`db.NOMECOLLEZIONE.find(SELETTORE)` → restituisce tutti i documenti, i cui campi rispettino la condizione espressa nella query.

`db.NOMECOLLEZIONE.find(SELETTORE, PROJECTION)` → restituisce tutti i campi `projection` dei documenti, i cui campi rispettino la condizione espressa nella query

Esempio di DB NoSQL: MongoDB

Esempi di utilizzo del costrutto di find

- **db.anagrafica.find()**
SELECT * FROM anagrafica
- **db.anagrafica.find({"nome": "Mario", "eta":30})**
SELECT * FROM anagrafica
WHERE ((Nome='Mario') AND (ETA=30))
- **db.anagrafica.find({"nome": "Mario"}, {"eta": 1})**
SELECT _ID,ETA FROM anagrafica
WHERE ((Nome='Mario')

Esempio di DB NoSQL: MongoDB

Esempi di utilizzo del costrutto di find

- `db.anagrafica.find({$or:[{"nome": "Mario"}, {"eta":56}]} , {eta:1})`
SELECT _ID, ETA
FROM anagrafica
WHERE ((Nome=Mario) OR (ETA=56))
- `db.anagrafica.find({"eta":{"$gte":60}})`
SELECT *
FROM ANAGRAFICA
WHERE (ETA >=60)

Esempio di DB NoSQL: MongoDB

Operatore di Ordinamento di una collezione

- `db.nomeCollezione.find(...).sort(CAMPO/CAMPI)`

1 = Ordinamento crescente, -1 = Ordinamento decrescente

```
db.anagrafica.find({"name": "Mario"}).sort({"eta":1})  
SELECT *  
FROM anagrafica  
WHERE (Name="Mario")  
ORDER BY ETA;
```

Esempio di DB NoSQL: MongoDB

Operatore di Conteggio di documenti

- `db.nomeCollezione.find(...).count()`

```
db.anagrafica.find({"nome": "Mario"}).count()  
SELECT COUNT(*)  
FROM anagrafica  
WHERE (Name="Mario")
```

Esempio di DB NoSQL: MongoDB

Operatori di Filtro di documenti duplicati

- `db.nomeCollezione.distinct([CAMPO],SELETTORE)`

```
db.anagrafica.distinct({"eta":1},{name:"Mario"})
```

```
SELECT DISTINCT(eta)
FROM anagrafica
WHERE (Name="Mario")
```

Esempio di DB NoSQL: MongoDB

- E' possibile raccogliere i comandi MongoDB in uno script (script in **linguaggio JavaScript**)

```
mongodb myfile.js
```

- La prima istruzione dello script contiene la connessione al server MongoDB, ed al database su cui si vuole operare:

```
conn = new Mongo();  
db = conn.getDB("tennis");
```

Esempio di DB NoSQL: MongoDB

Estensione Procedurale di MongoDB

- ✧ Il file di script può contenere **costrutti iterativi e/o di selezione**:

```
while(condizione) { LISTACOMANDI}  
  if(condizione) { LISTACOMANDI }  
  else { LISTACOMANDI }
```

- ✧ I **cursori** vengono usati per scorrere il risultato di una query.

```
cursor = db.collection.find(...);  
while ( cursor.hasNext() ) {  
    printjson( cursor.next() );  
}
```

Esempio di DB NoSQL: MongoDB

```
conn = new Mongo();
db = conn.getDB("tennis2");
db.createCollection("soci");
cursor = db.collection.find({"name"="mario"});
while (cursor.hasNext()) {
    printjson(cursor.next());
}
cursor = db.collection.find({"name"="mario"});
if (cursor.hasNext()) {
    print("Trovato!");
}
```

MYSCRIPT.js

Esempio di DB NoSQL: MongoDB

- Supponiamo di dover rappresentare **correlazioni tra collezioni** in MongoDB.

Es. Circoli Tennis e Soci dei Circoli


- MongoDB (come tutti i sistemi NoSQL) non mette a disposizione i costrutti di vincoli di integrità referenziale tra collezioni/tabelle.

Esempio di DB NoSQL: MongoDB

- Le **correlazioni** possono essere costruite esplicitamente mediante **campi** "replicati" tra più collezioni...

```
db.circoli.insert({"nome":"tennis2000",  
  "citta": "Bologna" })
```

```
db.soci.insert({"nome":"Mario",  
  "cognome":"Rossi", "nomeCircolo":"tennis2000"})
```



Esempio di DB NoSQL: MongoDB

- Le associazioni *uno-a-molti*, o *molti-a-molti*, tra documenti di diverse collezioni possono essere rappresentate sfruttando il fatto che in MongoDB il valore di un campo può essere anche un **array**, o una struttura complessa (es. documento annidato).

```
db.soci.insert({"name:'Mario', cognome:'Rossi',  
circolo:{ ... } })
```

Esempio di DB NoSQL: MongoDB

- Problema: come scrivere la query che restituisce nome e cognome dei soci che partecipano a circoli situati a Bologna, implementando il **JOIN tra collezioni**?

Soluzione 0: usando l'operatore di lookup (vedi prossime slide)

Soluzione 1: usando 2 query!

```
db.circoli.find({"luogo": "Bologna"}, {})
```

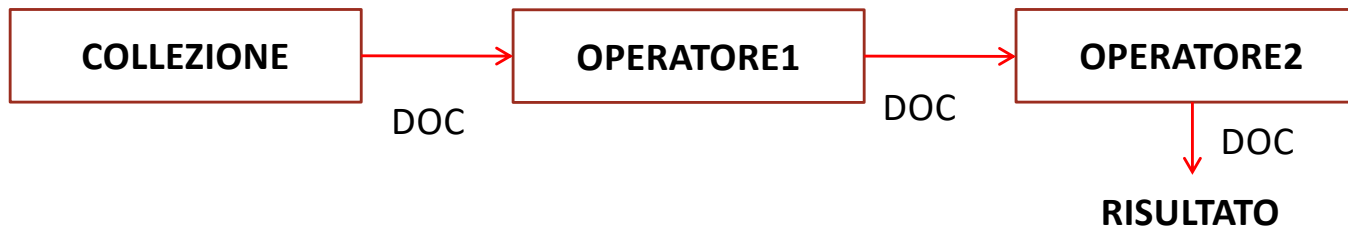
```
>> {"_id": "432"}
```

```
db.soci.find({"circolo": "432"},  
             {"nome": 1, "cognome": 1})
```

Esempio di DB NoSQL: MongoDB

Aggregazione dati mediante operatore di aggregate

- Consente di implementare una **pipeline** di operazioni da eseguire sulla base di dati.
- Ad ogni passo della pipeline, vengono eseguite operazioni che prendono in input dei documenti JSON e producono in output documenti JSON.



Esempio di DB NoSQL: MongoDB

ESEMPI DI OPERATORI

- `$geonear` → ordina i documenti dal più lontano al più vicino rispetto ad una posizione data
- `$match` → seleziona solo alcuni documenti che soddisfano le condizioni fornite in input
- `$project` → seleziona i campi prescelti
- `$group` → raggruppa in base ad uno o più campi
- `$limit` → seleziona i primi n documenti del JSON
- `$sort` → ordina il JSON in base ad alcuni campi
- `$out` → scrive l'output su una collezione

Esempio di DB NoSQL: MongoDB

ESEMPI DI OPERATORI

- **\$lookup** → consente di effettuare il **join** tra collezioni che appartengono allo stesso database

```
{ $lookup: {  
  from: collezione su cui fare il join,  
  localField: campo dalla collezione di partenza,  
  foreignField: campo della collezione del from,  
  as: nome del campo destinazione  
}
```

Esempio di DB NoSQL: MongoDB

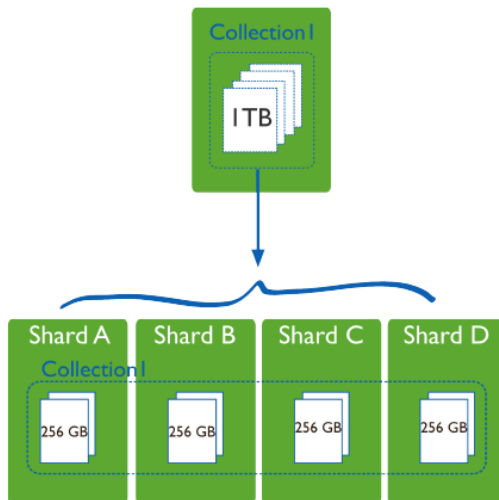
Aggregazione dati mediante operatore di aggregate

- `db.NOMECOLLEZIONE.aggregate([OPERATORE1, OPERATORE2, ..., OPERATOREN])`

```
db.anagrafica.aggregate([
    {$match: {name:'A'}},
    {$group: {_id:"$customId",
              total:{$sum: "$amount"}}
    ])
```

Esempio di DB NoSQL: MongoDB

- **Sharding** → Processo di *suddivisione dei dati* su un cluster di server MongoDB (database distribuito).



Perché distribuire i dati?

- Maggiore capacità di storage
- Maggiore capacità computazionale
- Ridondanza dei dati
- Disponibilità del servizio
- ...

Esempio di DB NoSQL: MongoDB

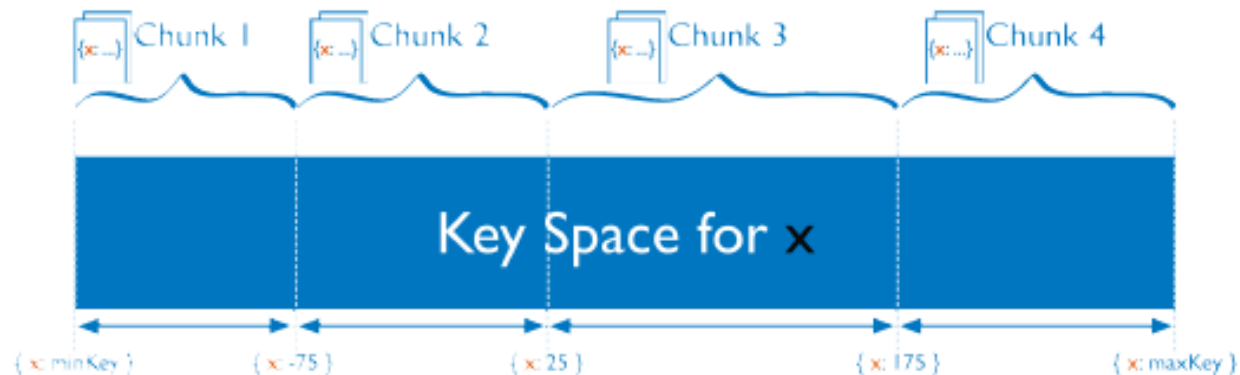
D. Come effettuare la suddivisione dei dati in un sistema distribuito?

- Necessità di definire una **shard key**.
- La chiave **deve essere presente su tutti i documenti di una collezione!**
- In base al valore della chiave, si suddivide la collezione in **segmenti** (chunks).
- Gruppi di chunk vengono assegnati ai diversi nodi del cluster
Come effettuare l'allocazione?

Esempio di DB NoSQL: MongoDB

(SOLUZIONE1) RANGE-BASED SHARDING

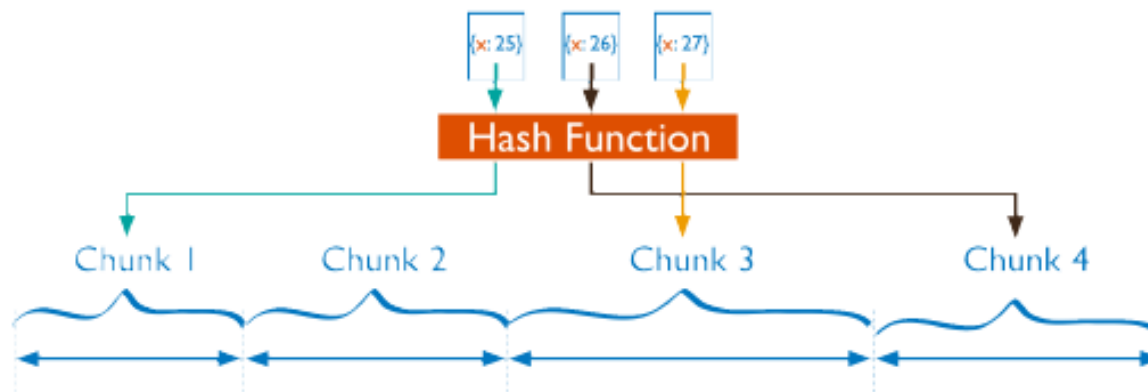
- Individua valore massimo e minimo della chiave
- Ogni **chunk** corrisponde ad un intervallo $[K-i, K+i]$



Esempio di DB NoSQL: MongoDB

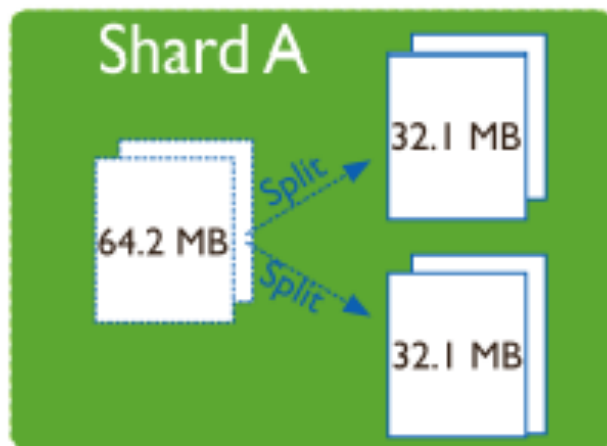
(SOLUZIONE2) HASH-BASED SHARDING

- MongoDB applica la funzione hash(#chunk)
- Il risultato della funzione determina il server.



Esempio di DB NoSQL: MongoDB

- Ottimizzazioni a run-time: **Splitting**



- Se un chunk cresce troppo in dimensione, esso viene splittato in più parti.
- L'operazione viene eseguita su un server, e non comprende la migrazione del chunk stesso.

Esempio di DB NoSQL: MongoDB

- Ottimizzazioni a run-time: **Balancing**
- Il **Balancer** viene eseguito in **background** e tiene traccia del numero di chunk gestito da ciascun server.
- In caso di allocazione non bilanciata, il Balancer provvede a **migrare i chunk** tra server differenti (dal più carico al più scarico).

