

No SQL Documental Con MongoDB

Ing. Sergio Méndez
IT Manager/Fundador

FLISOL Cobán

Agenda

- Big Data
- Bases de datos No SQL
- Tipos de DB No SQL
- Ejemplos
- mongoDB
- JSON

Agenda

- Colecciones
- Funcionamiento de Mongo
- CRUD en mongoDB
- Embebido versus Normalización
- Tipos de relaciones
- Bases de datos Polimórfica
- Autobalance

Agenda

- Sharding
- Autobalanceo
- GridFS
- Aplicaciones
- Demostración
- Preguntas



Big Data

Big Data es en el sector de tecnologías de la información y la comunicación una referencia a los sistemas que manipulan grandes conjuntos de datos (o data sets). Las dificultades más habituales en estos casos se centran en la captura, el almacenamiento, búsqueda, compartición, análisis y visualización.

Cómo manejo Big Data?

Not only SQL

Bases de datos NoSQL

Definición: Son la siguiente generación de bases de datos que apuntan a ciertos puntos: ser no relacionales, distribuidas, de código abierto y ser horizontalmente escalables. La intención original ha sido las bases de datos web escalables.

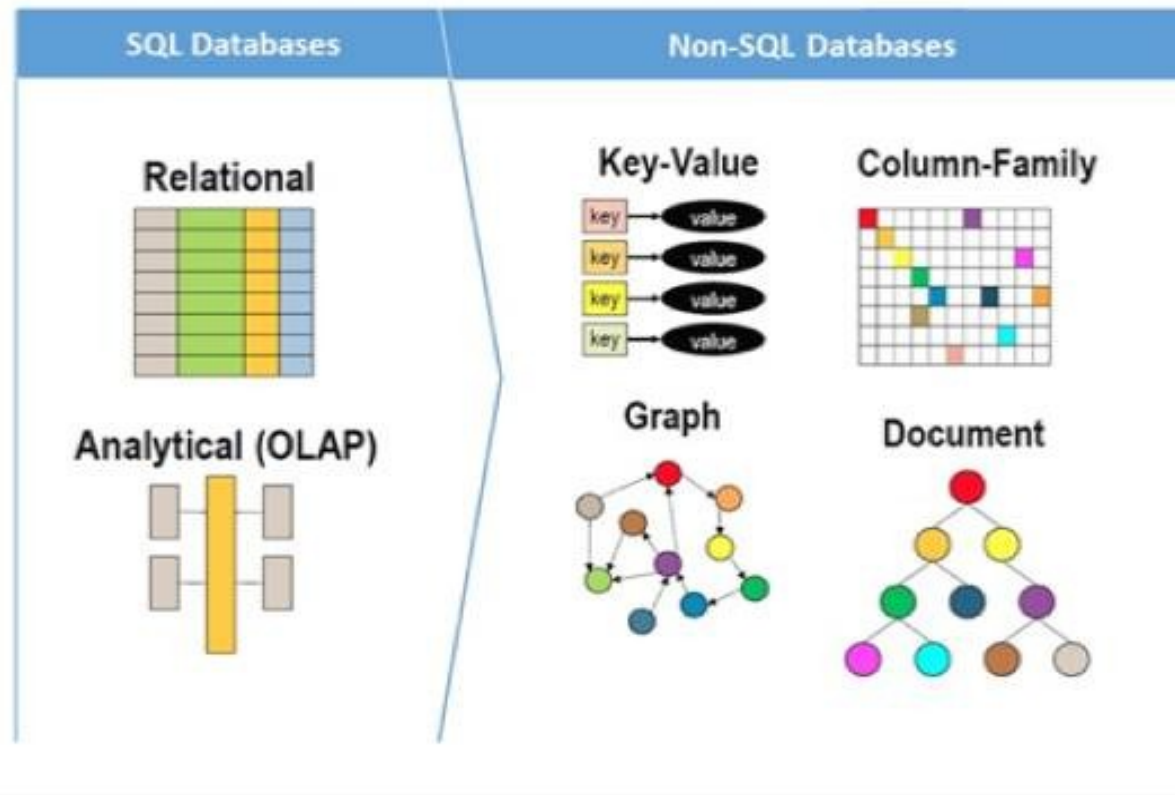
Bases de datos NoSQL

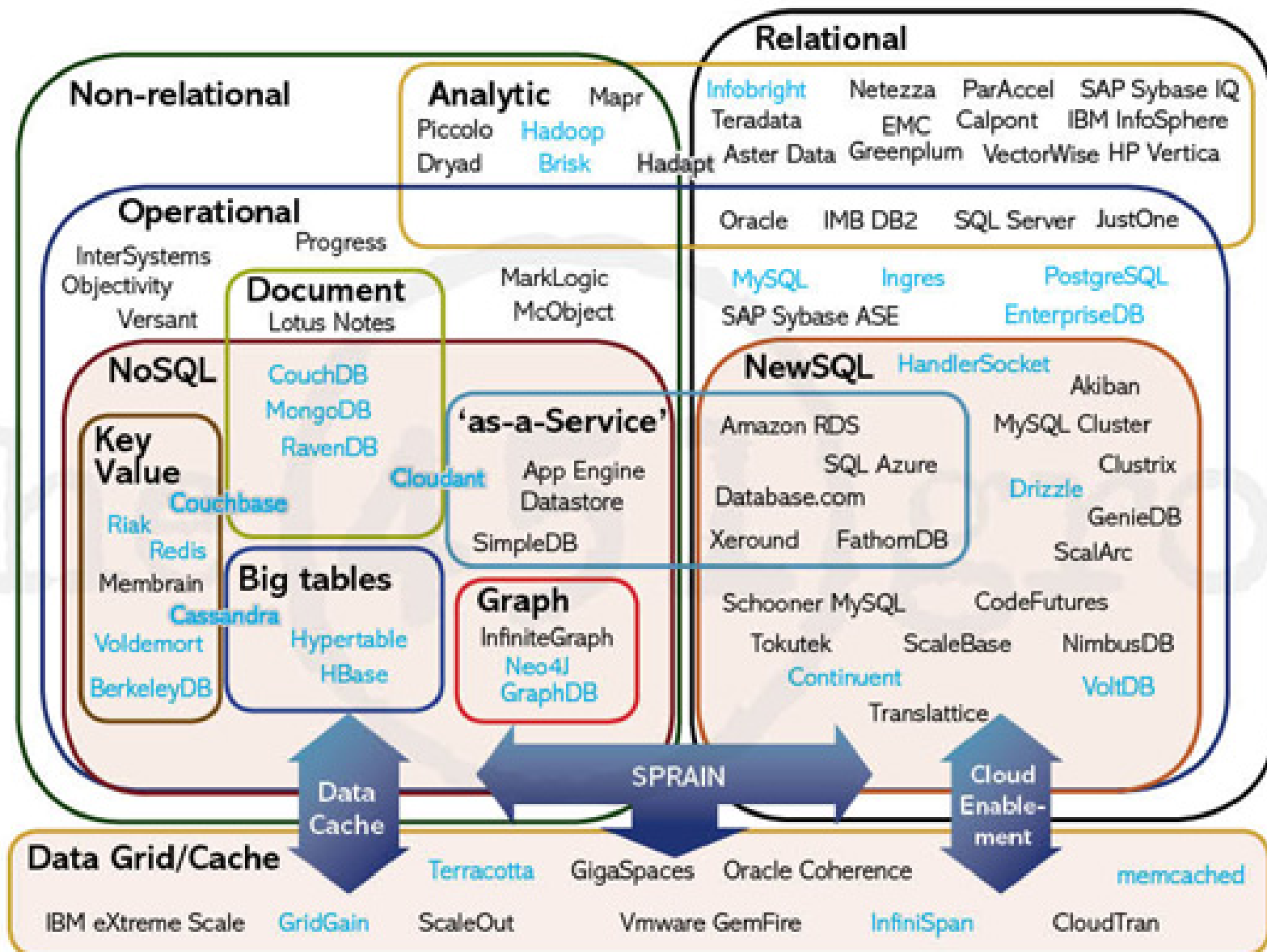
El movimiento comenzó en el año 2009 y ha ido creciendo rápidamente. Otras características aplican como ser libres de esquemas, con soporte sencillo para replicación con API simple, eventualmente consistente o que no cumpla con las características ACID.

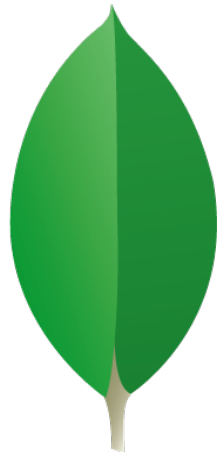
El termino “nosql” se refiere a “not only sql” puede verse como un alias de la definición anterior.

Tipos de base de datos No SQL

noSQL: “Not Only SQL”







mongoDB

MongoDB

MongoDB (proviene de "humongous") es una base de datos NoSQL escalable, de alto rendimiento y de código abierto

Características:

- Almacenamiento orientado a documentos
- Soporte completo de Indexado
- Replicación y alta disponibilidad
- Auto Sharding
- Consultas
- Fácil actualización
- Map/Reduce
- GridFS
- Soporte Comercial

{JSON}

JSON

JSON, acrónimo de JavaScript Object Notation, es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML.

Almacenando con JSON

MongoDB almacena todos los documentos en estructuras compuestas en formato JSON compuestas por pares de campos y valores:

```
{ "item": "pencil", "qty": 500, "type": "no.2" }
```

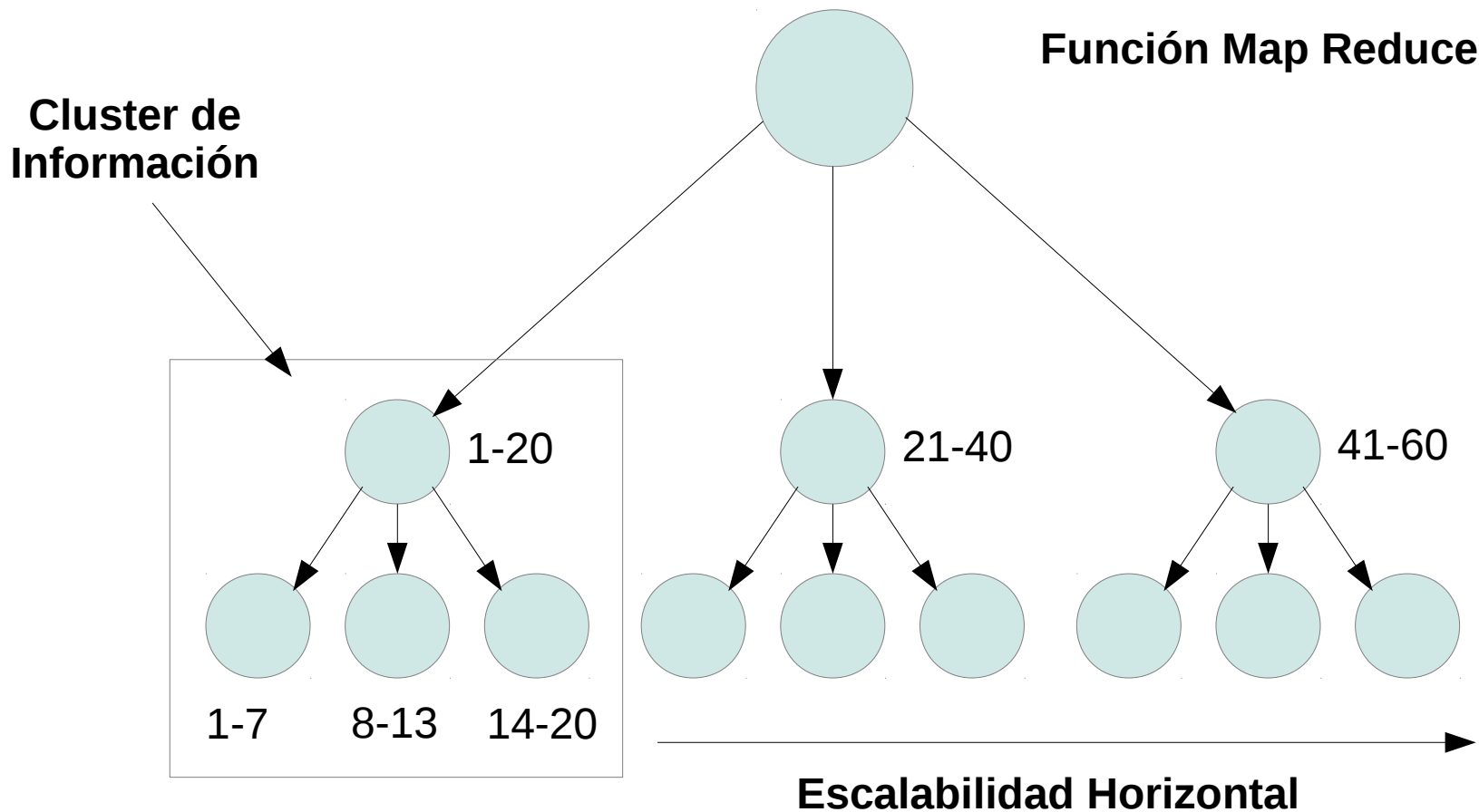
Colecciones

Una colección define el grupo donde se almacenan los documentos dentro de la base de datos de MongoDB.

Funcionamiento Interno

Map: Ubicar el nodo que tiene la información

Reduce: Obtener y devolver la información del nodo ubicado por la función Map



CRUD en mongoDB

Create

```
db.collection.insert( { data } );
```

Read

```
db.collection.find( { query } );
```

Update

```
db.collection.update( {query} , { data } );
```

Delete

```
db.collection.remove( { query } );
```

Embebido versus Normalización

Tipos de relaciones

Relación de uno a uno

Normalizado

```
{
  _id: "joe",
  name: "Joe Bookreader"
}

{
  patron_id: "joe",
  street: "123 Fake Street",
  city: "Faketon",
  state: "MA",
  zip: "12345"
}
```

Embebido

```
{
  _id: "joe",
  name: "Joe Bookreader",
  address: {
    street: "123 Fake
Street",
    city: "Faketon",
    state: "MA",
    zip: "12345"
  }
}
```


Relación de uno a muchos

Normalizado

```
{
  _id: "joe",
  name: "Joe Bookreader"
}

{
  patron_id: "joe",
  street: "123 Fake Street",
  city: "Faketon",
  state: "MA",
  zip: "12345"
}

{
  patron_id: "joe",
  street: "1 Some Other Street",
  city: "Boston",
  state: "MA",
  zip: "12345"
}
```

Embebido

```
{
  _id: "joe",
  name: "Joe Bookreader",
  addresses: [
    {
      street: "123 Fake Street",
      city: "Faketon",
      state: "MA",
      zip: "12345"
    },
    {
      street: "1 Some Other Street",
      city: "Boston",
      state: "MA",
      zip: "12345"
    }
  ]
}
```

Relación de muchos a muchos

Colección de Categorías

```
{  
  _id: "juguetes",  
  productos: [1001,1002,1003,1004]  
}
```

Colección de Productos

```
{  
  _id: "1001",  
  nombre: "Minion",  
  cantidad: 3,  
  categoria: ["juguetes"],  
  precio: "50.00"  
}
```

Base de datos Polimórfica

Polimorfismo

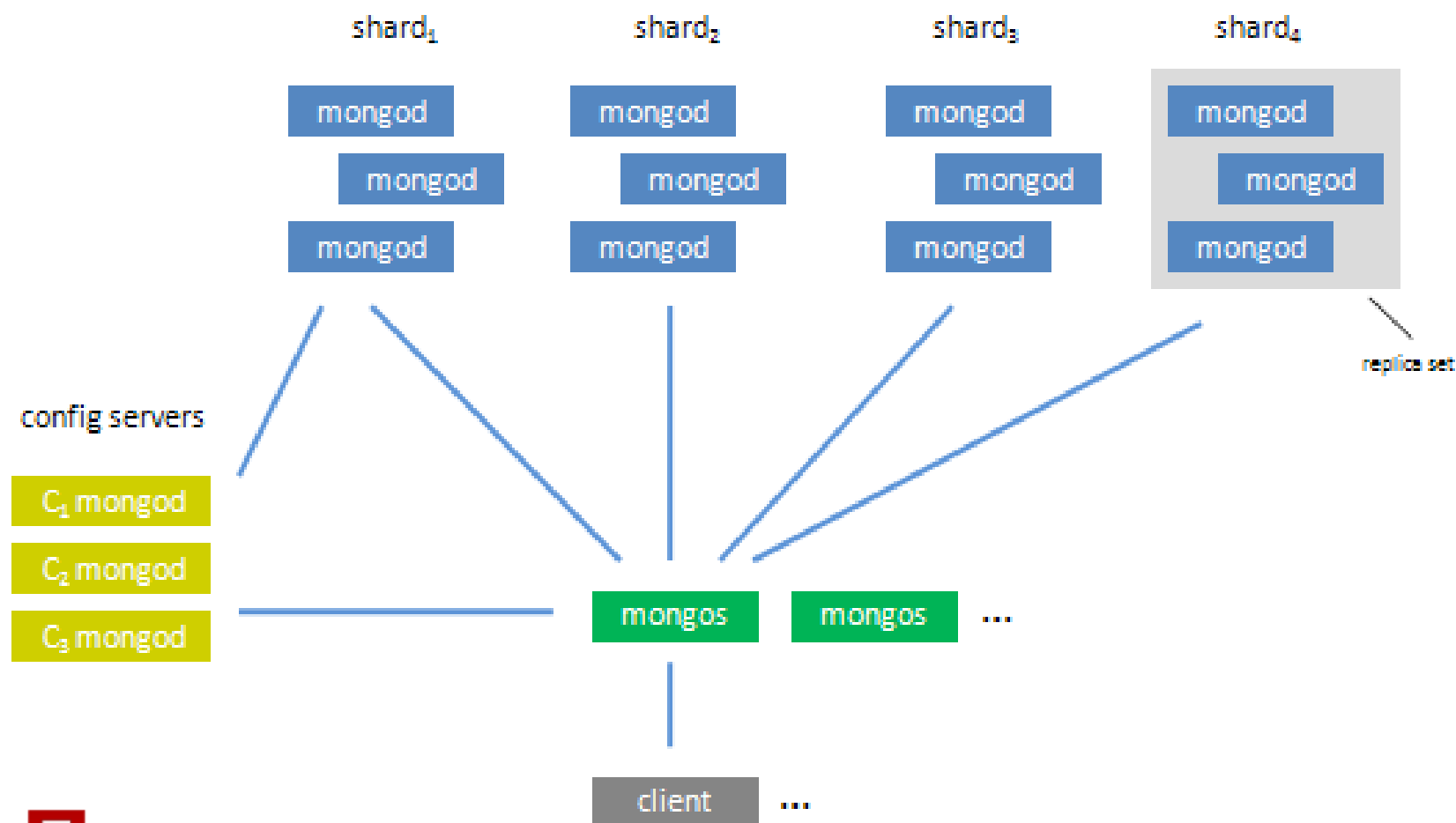
Especifica que una acción puede tener el mismo nombre en diferentes clases y cada clase ejecutará tal operación de forma distinta. En este caso una documento puede llamarse igual pero tener distinto contenido.

```
db.servivo.insert( { nombre: 'pez', aletas:2 } );  
db.servivo.insert( { nombre: 'perro', aullido: 'fuerte' } );  
db.servivo.insert( { nombre: 'culebra', veneno: 'fuerte' } );  
db.servivo.insert( { nombre: 'araña', patas: 8 } );
```

```
db.servivo.find();
```

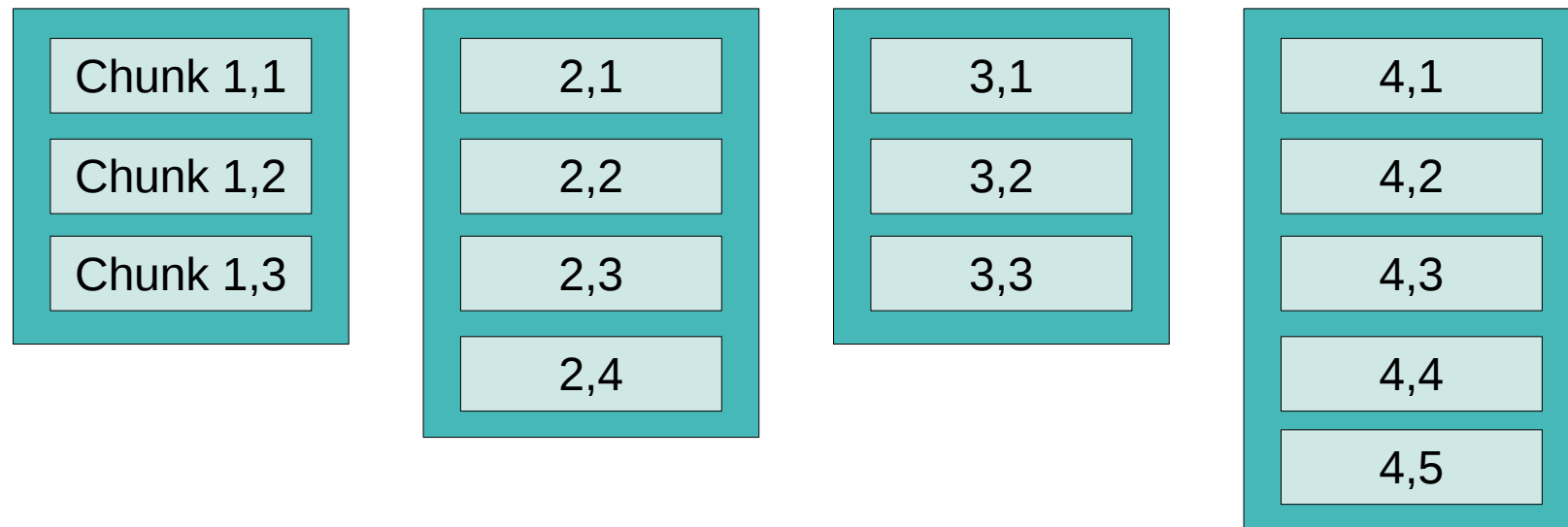
Sharding

Es una técnica de particionamiento que provee escalabilidad horizontal.



Autobalanceo

Técnica para distribuir los datos equitativamente entre nodos para generar una carga similar entre nodos de información



Funcionamiento de GridFS

GridFS es una especificación para almacenamiento y recuperación de archivos que exceden el límite de 16MB en formato BSON-document.

Con GridFS se almacenan los archivos en varias partes o chunks y almacenan cada chunk como un documento separado. Por defecto en GridFS se limita cada chunk a 255k.

Cuando consultas un archivo almacenado en GridFS el driver o el cliente que lo consulta hace que el archivo reensamble todos los chunks que necesite. Esto permite que también se pueda saltar chunks en cualquier parte del archivo.

Colecciones de GridFS

Los archivos de GridFS se almacenan en 2 colecciones:

chunks almacena los chunks binarios de cada archivo

files almacena metadatos para el archivo

Estos son almacenados en las colecciones GridFS dentro del bloque común usando el prefijo fs.

fs.files

fs.chunks

Aplicaciones en Producción

Perfecto para aplicaciones de:

- Manejo de archivos y logs
- Administración de documentos
- CMS o contenido web
- Ecommerce
- Juegos
- Estadísticas en tiempo real
- Analíticas



No recomendable para proyectos orientados a transacciones, como bancos y manejo de cuentas

Demostración

Información de Contacto

Sitio web



<http://www.innovaos.com>

Síguenos en



<http://www.facebook.com/innovafanpage>



<http://www.twitter.com/innovaos>



<http://gt.linkedin.com/in/innovaos>

Escribenos a



sergioarm@innovaos.com

o llámanos a los números



(+502) 24336665 - 54401554