

Universidade Federal de Goiás
Ciências da Computação
Sistemas Operacionais 2

MongoDB

Diego Fraga - 083746
Gabriel Henrique - 083771
Lauro Henrique - 074259



O que é MongoDB?

- MongoDB é uma ferramenta de banco de dados open source, escalável, de alto desempenho e escrita em C.
- Muito diferente dos bancos de dados relacionais, já que eles não armazenam dados em tabelas e sim na forma de documentos.



Para que serve o MongoDB?

- Ele foi projetado para problemas que não são facilmente resolvidos por RDBMSs (sistema de gerenciamento de banco de dados relacional) tradicionais.
- Ele gerencia coleções de documentos. Isso permite que muitas aplicações sejam mais naturais, pois os dados podem ser aninhados em hierarquias complexas.



Por que escolher um armazenamento sem esquemas?

- Sob a perspectiva de um desenvolvedor, dados orientados a documento (ou sem esquema) são mais fáceis e muito mais flexíveis de se gerenciar do que os dados relacionais. Os documentos são gravados individualmente, contendo qualquer dado necessário.
- Porém, o armazenamento sem esquemas não é apropriado para todo domínio.



Por que escolher um armazenamento sem esquemas?

- Modelar um exemplo de uma Cartão de visitas em termos relacionais é possível, mas complicado. Um banco de dados relacional, acaba tendo muitos registros com um valor nulo em algumas colunas, para cada um ou dois registros que usam esse valor. Também é preciso especificar tipos de colunas em um sistema relacional, de modo que você possa ficar limitado, digamos, pelo tamanho do campo de endereço.



Características do MongoDB

- Suporte multi-plataforma: os binários estão disponíveis para Windows, Linux, OS X e Solaris.
- Suporta tipos ricos: datas, expressões regulares, códigos, dados binários, outros.
- A velocidade é outra vantagem do Mongo DB, principalmente em relação a como ele trata as gravações: elas são armazenadas na memória e mais tarde, através de encadeamento de segundo plano, gravadas em disco.



Características do MongoDB

- A API do MongoDB é uma mistura nativa de objetos JSON e funções de JavaScript. Os desenvolvedores interagem com o MongoDB através do programa de shell, que permite argumentos de linha de comandos.
- Duas das principais características do MongoDB são as funções: MapReduce e Sharding.



MapReduce

- A filosofia por trás do MapReduce é: Diferentemente de data-stores centrais, como um banco de dados, você não pode assumir que todos os dados residem em um lugar central portanto você não pode executar uma query e esperar obter os resultados em uma operação síncrona. Em vez disso, você precisa executar a query em cada fonte de dados simultaneamente.
- O processo de mapear a requisição do originador para o data source é chamado de 'Map', e o processo de agregação do resultado em um resultado consolidado é chamado de 'Reduce'.

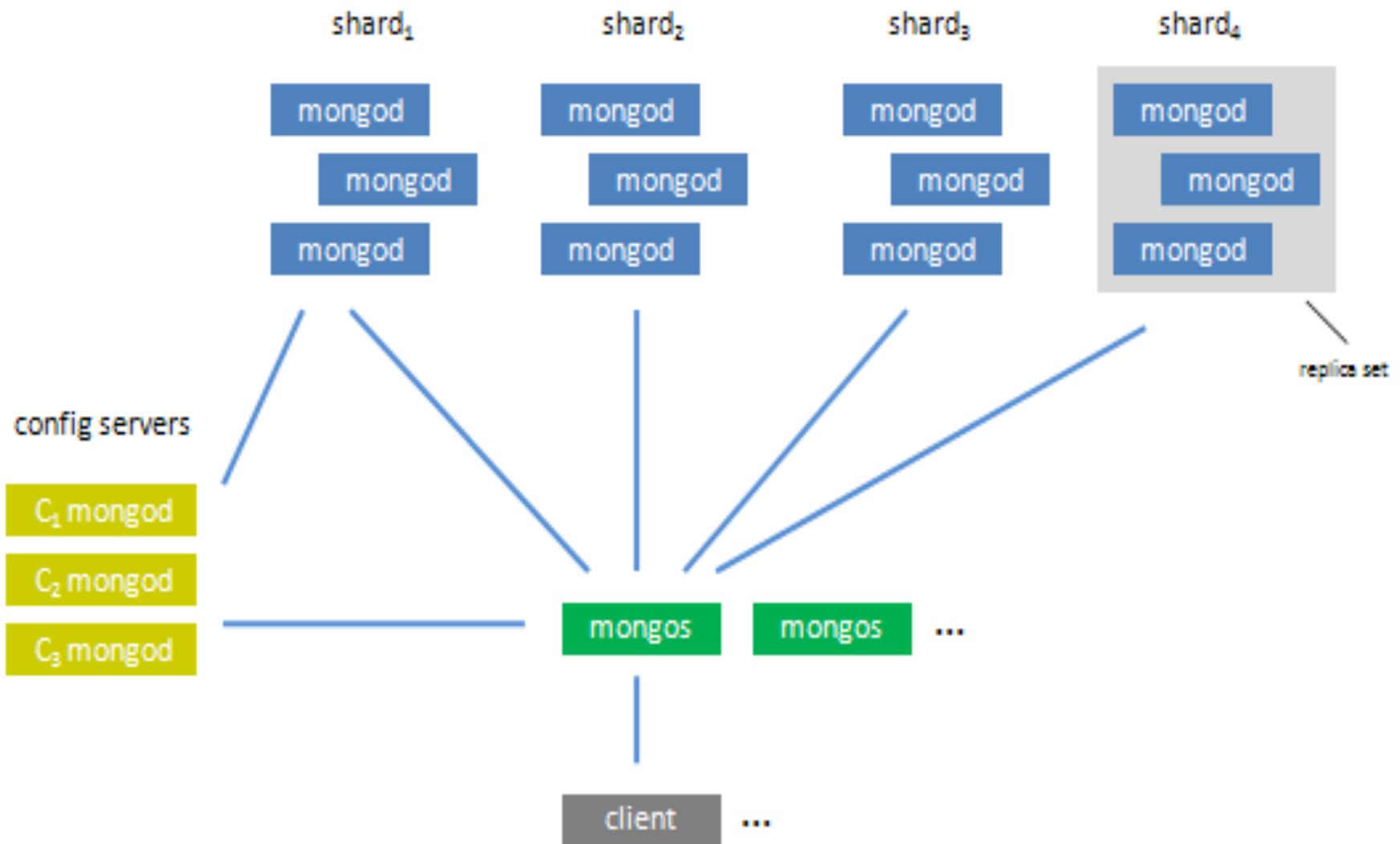


Sharding

- Sharding é uma técnica muito usada atualmente para lidar com escalabilidade de massas de dados, consiste basicamente em dividir os dados de uma aplicação entre vários bancos: por exemplo, numa aplicação com 1000 usuários o sharding faria com que os usuários com nomes de A a J ficassem em um servidor, e de K a Z em outro servidor.
- Se você tem um monte de dados e você está no limite de disco e / ou falta de espaço, uma maneira de tratar isso utilizando sharding é ter os seus dados divididos entre várias máquinas.



Sharding



Contexto: qual é o estado deste tipo de tecnologia atualmente?



Quem utiliza/exemplos de possíveis utilizações

Hoje é usado por pelo menos 40 sites de alto acesso e performance, incluindo sites como *SourceForge*, *GitHub*, *EA* e *The New York Times*, *Google*, *Twitter*



Query

```
BasicDBObject doc = new BasicDBObject();
```

```
doc.put("name", "MongoDB"); BasicDBObject info = new  
BasicDBObject();
```

```
info.put("x", 203);
```

```
doc.put("info", info);
```

```
coll.insert(doc);
```



Query

```
query = new BasicDBObject();
```

```
query.put("i",... new BasicDBObject("$gt", 50));
```

```
cur = coll.find(query);while(cur.hasNext()) {
```

```
    System.out.println(cur.next());
```

```
}
```



SQL x NoSQL (MongoDB)

INSERT INTO USERS VALUES(1,1)

```
db.users.insert({a:1,b:1})
```

UPDATE users SET a=1 WHERE b='q'

```
db.users.update({b:'q'}, {$set:{a:1}}, false, true)
```



SQL x NoSQL (MongoDB)

```
SELECT COUNT(*) FROM users  
db.users.count()
```

```
SELECT a,b FROM users db.users.find({}, {a:1,b:1})
```



Demonstração

<http://www.mongodb.org/display/DOCS/Tutorial>

<http://hi.baidu.com/hxzon/blog/item/b827af771b90ac16b151b95e.html>



Demonstração

Criando um Documento com vários campos:

```
Kico = {  
  nome: "Henrique Lobo Weissmann",  
  apelido: "Kico (ou seria este o nome?)",  
  cidade: "Belo Horizonte"  
}
```

Armazenando num Banco de Dados cujo nome é:
"kicodb"

```
Cdb.kicodb.save(Kico)
```



Vantagens x Desvantagens

Vantagens:

- Finalmente você se livrou de uma abordagem bidimensional e pode representar objetos do mundo real como realmente são: complexos e únicos.
- Caso no futuro surja algum caso no qual novos atributos apareçam, você pode aplicá-los somente aonde é necessário, e não em todos os casos, como no modelo relacional, aonde normalmente cria-se uma nova “coluna” na tabela relacionada



Bibliografia

<http://www.mongodb.org/>

<http://en.wikipedia.org/wiki/MongoDB>

<http://www.nosqlbr.com.br/>

<http://www.ibm.com/developerworks/br/java/library/j-javadev2-12/index.html>

<http://core.eti.br/>

<http://www.zorched.net/2010/10/04/mongodb-mapreduce-functions-for-grouping/>

