

Banco de Dados NoSQL

(NS0201)

Prof. Giovane Barcelos
giovane_barcelos@uniritter.edu.br

NoSQL - Redis

Conteúdo Programático

- 1. Introdução ao Redis**
- 2. Primeiros Passos**
- 3. Tipos de Dados**
- 4. Persistência de Dados**
- 5. Aplicação prática com Node.js**

1. Introdução ao Redis

Quais serão os tópicos que serão cobertos?

- **Fundamentos do Redis**
- **Modelo baseado em Memória vs Disco**
- **Estrutura de dados**
- **Segurança & autenticação**



1. Introdução ao Redis

O que é o Redis?

- Redis é uma **estrutura de dados** armazenada em **memória de código livre** (*opensource*) que persiste em disco, mas mantém 100% do banco em memória
- Usada como **banco de dados** que atua como **cache** e intermediador (*broker*) de **mensagens**
- É um **NoSql** com **armazenamento** do tipo **Chave/Valor** (*Key/Value*)

Key	Value
K1	AAA,BBB,CCC
K2	AAA,BBB
K3	AAA,DDD
K4	AAA,2,01/01/2015
K5	3,ZZZ,5623

1. Introdução ao Redis

Como *Redis* se relaciona com *MongoDB*?

- **Redis** é **semelhante** ao **MongoDB** na medida em que é **também** um **NoSQL** e pode ser usado para projetos que necessitem de **chave/valor**
- **MongoDB**, no entanto, é **classificado** como **NoSQL** do tipo **documento** e **persistido** em **disco** enquanto o **Redis** é **apenas chave/valor** e baseado **em memória**
- **Redis** é usado para **armazenamento** de **cache**



1. Introdução ao Redis

Ele é similar ao Memcached?

- **Redis** é similar ao **Memcached** porque ele também armazena os dados em **memória**
- **Memcached** usa **cache volátil**, portanto seus **dados** não são persistentes e são **perdidos** com o **reinício** do serviço
- **Redis** usa **persistência embutida** e não perderá os **dados** com o **reinício** do serviço

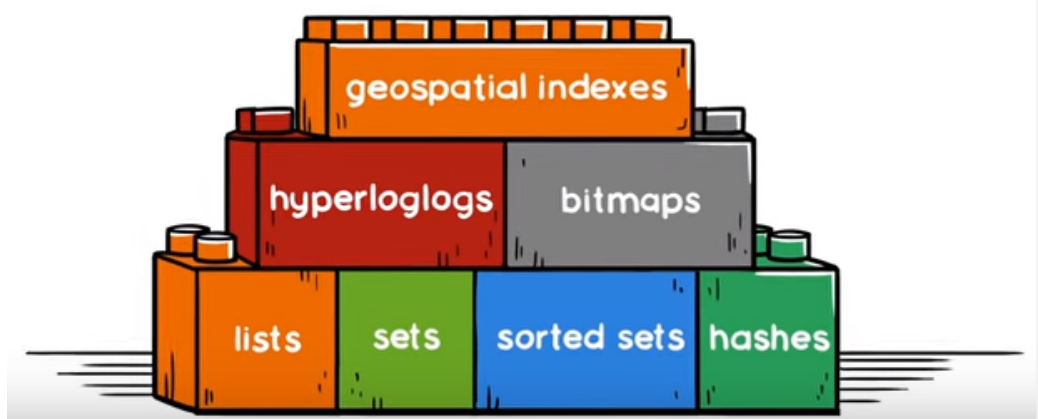
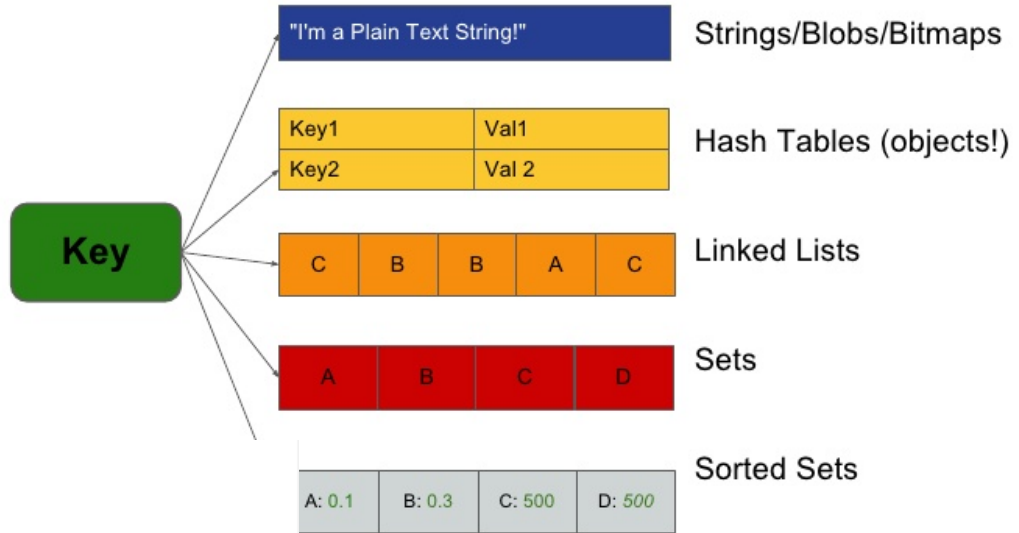


1. Introdução ao Redis

Quais são só tipos de dados que o *Redis* suporta?

- 1. Strings
- 2. Hashes
- 3. Lists
- 4. Sets
- 5. Sorted Sets
- 6. Bitmaps
- 7. Hyperloglogs
- 8. Geospatial Indexes

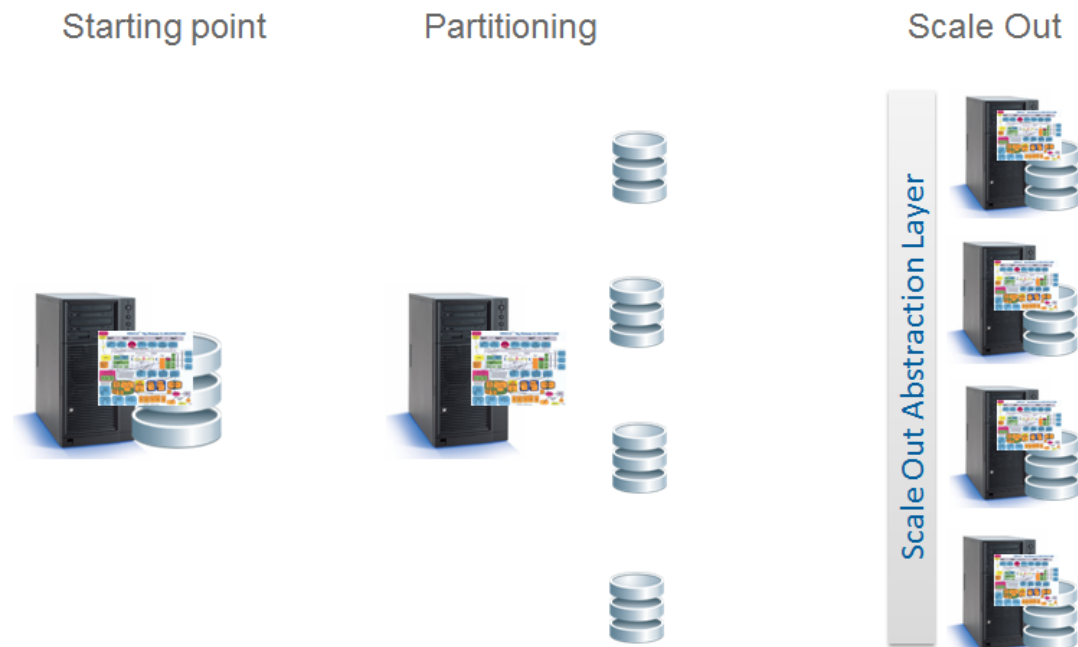
Key => { Data Structures }



1. Introdução ao Redis

Redis escala (*scaling*) & particiona (*partitioning*)?

- Estas técnicas não são facilmente configuradas no *Redis*
- Depende se o uso é para *cache* ou *armazenamento de dados*
- **Particionamento (*partitioning*)**: dados distribuídos em várias instâncias de *Redis*



1. Introdução ao Redis

Qual é o modelo de segurança do Redis?

- **Redis** foi **projetado** para ser **acessado** por **clientes confiáveis** dentro de **ambientes confiáveis**
- Não é uma boa ideia expor o *Redis* diretamente
- Os **clientes** no mesmo ambiente **podem acessar o Redis via web**
- **Não otimizado** para **segurança máxima**, mas para **desempenho máximo**



1. Introdução ao Redis

Quais são os clientes do Redis?

- **Redis** pode acessado por praticamente qualquer linguagem popular (<http://redis.io/clients>)

ActionScript	Bash	C	C#	C++	Clojure
Common Lisp	Crystal	D	Dart	Delphi	Elixir
emacs lisp	Erlang	Fancy	gawk	GNU Prolog	Go
Haskell	Haxe	Io	Java	Julia	Lasso
Lua	Matlab	mruby	Nim	Node.js	Objective-C
OCaml	Pascal	Perl	PHP	PL/SQL	Pure Data
Python	R	Racket	Rebol	Ruby	Rust
Scala	Scheme	Smalltalk	Swift	Tcl	VB
VCL					

1. Introdução ao Redis

Como é a Estrutura de Dados do Servidor?

- ***Redis não possui dados brutos no servidor***
- ***Não possui esquemas e nomes de colunas***
- ***Como os dados são representados? *Sets, hashes, etc****
- ***Leva vantagem na velocidade***

1. Introdução ao Redis

Qual a diferença em relação aos SGBDs tradicionais?

- Tradicionalmente bancos de dados **pesquisam** uma **tabela inteira** ou pesquisam em um **índice**
- **Redis** possui comandos de **recuperação direta de dados**
- Não possui um motor de pesquisa interno
- Deve-se tomar uma decisão sobre como armazenar e retornar dos dados



1. Introdução ao Redis

Algumas questões sobre o retorno dos dados

- **Os valores precisam de chaves?**
- **Existem múltiplos campos para uma única chave?**
- **São contados elementos em uma coleção?**
- **Armazena objetos?**
- **Possível ordenar de forma crescente / decrescente?**
- **Tem valores únicos?**
- **Testa se os valores existem?**

Strings, Sets, Stored Sets, Hashes & Lists

1. Introdução ao Redis

Indexação Customizada

- **Leitura de dados por mais de uma chave de recuperação**

User:[UserId]

User:[Name]

GET UserName:[Name]

1. Introdução ao Redis

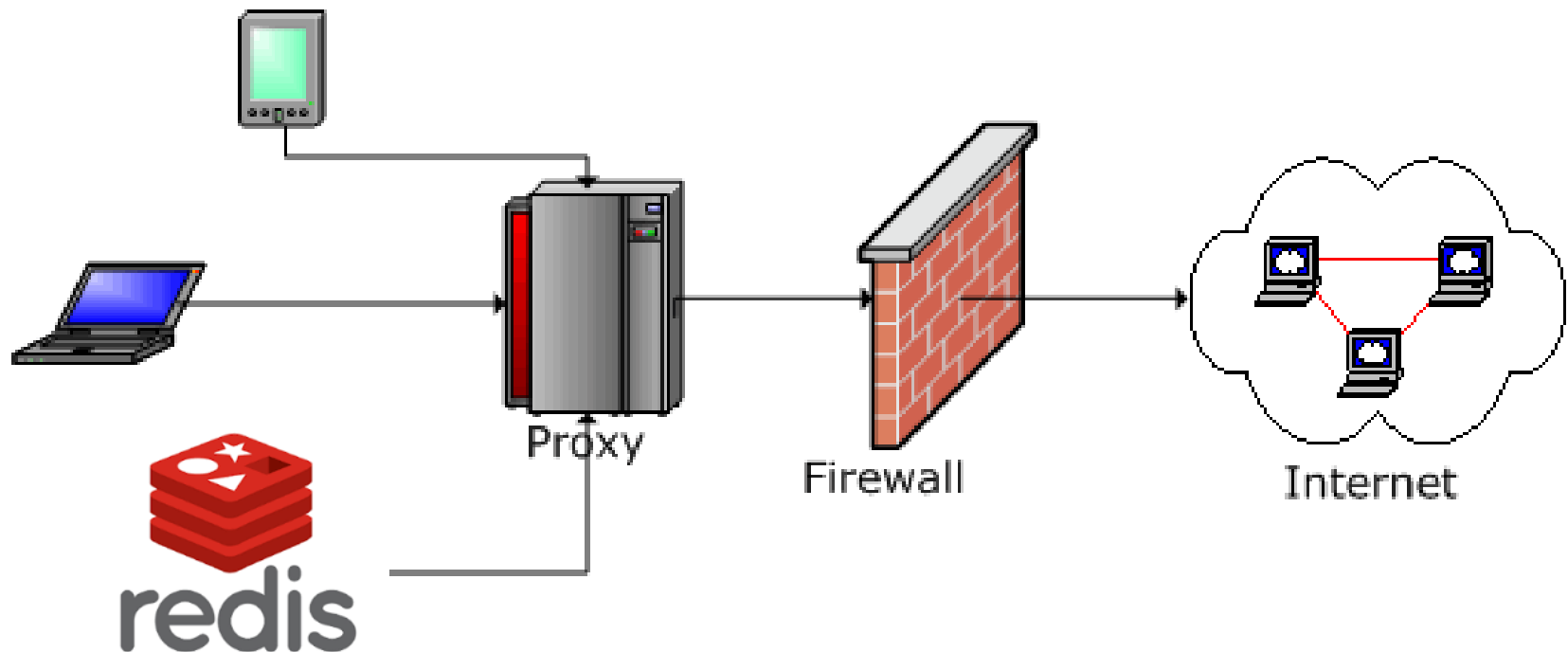
E o armazenamento dos dados?

- Diferente do **SQL**
- Armazena os dados que o desenvolver **CONHECE** e irá usar
- A leitura é explícita e **NÃO descoberta** aleatória
- Os dados são **armazenados em memória** em tempo de execução

1. Introdução ao Redis

Qual é o modelo de segurança?

- Projetado para ser **acessado** por **clientes confiáveis**
- Se possível **não permita** nenhum **acesso externo**
- A **autenticação simples** pode ser configurada
- **Redis** pode ser **restrito** a certas **interfaces**



1. Introdução ao Redis

E a segurança de rede?

- Deve-se **negar acesso** à **porta principal** de *Redis*
- Configure um **firewall** para impedir o acesso
- **Loopback** é uma alternativa para virtualizar o acesso
- Pode-se ligar *Redis* a uma **única interface** (**redis.conf**): **Bind 127.0.0.1**

1. Introdução ao Redis

Autenticação

- Possui uma **pequena camada** de autenticação (habilitar em **redis.conf**)
- **Recusa** todas as consultas de **usuários não autenticados**
- O usuário deve enviar o **comando AUTH** junto com a **senha**
- A **senha** é armazenada em **texto simples / aberto** no arquivo **redis.conf**

1. Introdução ao Redis

Encrytação de Dados

- ***Redis não*** suporta **criptografia** de dados
- Se o ***Redis*** estiver aberto para ***web***, use **camada de proteção adicional**
- ***SSL Proxy - Spiped***

1. Introdução ao Redis

Como desabilitar e renomear comandos?

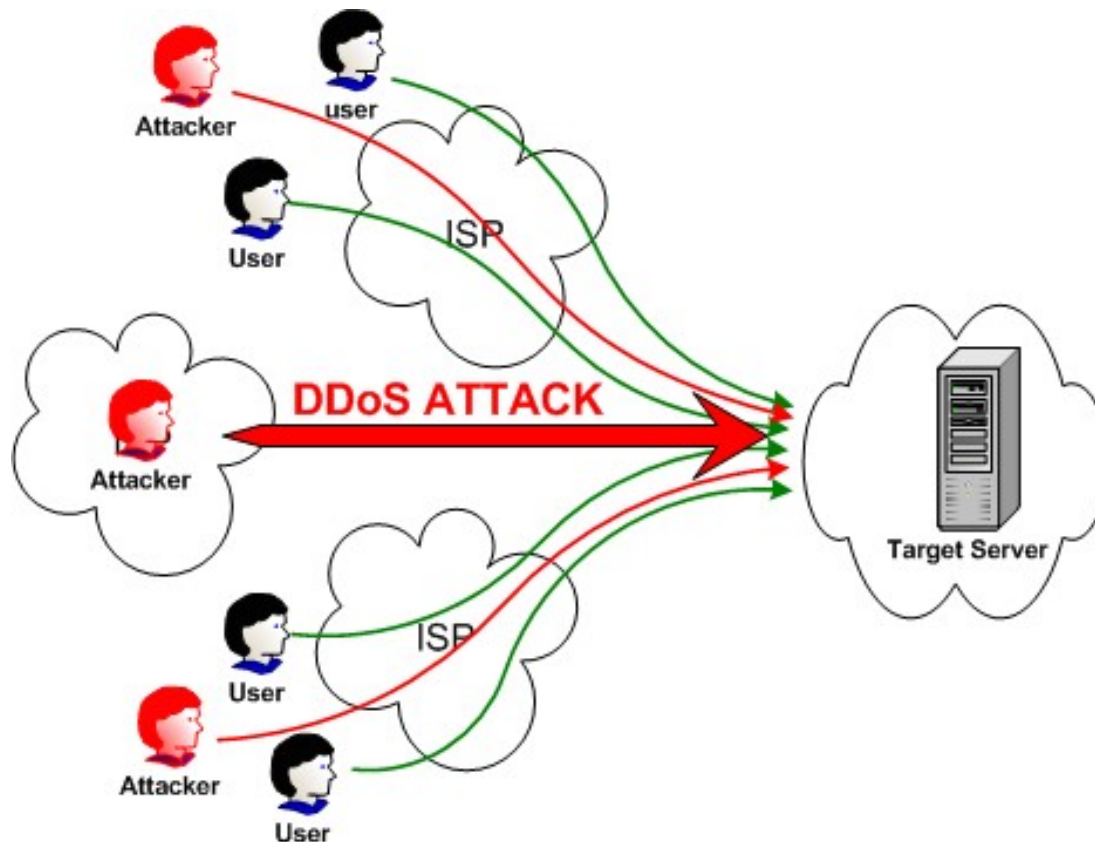
- **Comandos** específicos **podem** ser **desabilitados** ou **renomeados**
- Os **usuários normais não** podem ser capazes de executar ***CONFIG*** ou ***FLUSHALL***
- Para desabilitar, renomeie para *string* vazia
- Os comandos são renomeados em **redis.conf**
rename-command CONFIG configuracao

rename-command CONFIG ""

1. Introdução ao Redis

Vulnerabilidades de ataques

- Os ataques podem ser disparados de fora, mesmo se o *Redis* esteja seguro
- Vulnerabilidade pela *Web*: ataque *DoS*



Lembre-se

“Que a força esteja contigo.”

Star Wars

