第四章

NoSQL

—— MongoDB

—— Redis

内容提要

- 在 Spring 中访问 MongoDB
- 在 Spring 中访问 Redis
- Spring 的缓存抽象
- 基于Redis的缓存

非关系型数据库-MongoDB

- 属于文档型数据库，可以存放xml、json、bson类型的数据。这些数据具备自述性，呈现分层的树状数据结构。数据结构由键值(key=>value)对组成。
- 存储方式：虚拟内存+持久化。
- 查询语句：是独特的MongoDB的查询方式。
- 适合场景：事件的记录，内容管理或者博客平台等等。
- 架构特点：可以通过副本集，以及分片来实现高可用。

MongoDB优势与劣势

- **优势**
  - 在适量级的内存的**MongoDB**的性能是非常迅速的，它将热数据存储在物理内存中，使得热数据的读写变得十分快。
  - **MongoDB**的高可用和集群架构拥有十分高的扩展性。
  - 在副本集中，当主库遇到问题，无法继续提供服务的时候，副本集将选举一个新的主库继续提供服务。
  - **MongoDB**的**Bson**和**JSon**格式的数据十分适合文档格式的存储与查询。
- **劣势**
  - 不支持事务操作。**MongoDB**本身没有自带事务机制，若需要在**MongoDB**中实现事务机制，需通过一个额外的表，从逻辑上自行实现事务。
  - 应用经验少，由于**NoSQL**兴起时间短，应用经验相比关系型数据库较少。
  - **MongoDB**占用空间过大。

MongoDB vs MySQL

数据库	MongoDB	MySQL
数据库模型	非关系型	关系型
存储方式	以类JSON的文档的格式存储	不同引擎有不同的存储方式
查询语句	MongoDB查询方式（类似JavaScript的函数）	SQL语句
数据处理方式	基于内存，将热数据存放在物理内存中，从而达到高速读写	不同引擎有自己的特点
成熟度	新兴数据库，成熟度较低	成熟度高
广泛度	NoSQL数据库中，比较完善且开源，使用人数在不断增长	开源数据库，市场份额不断增长
事务性	仅支持单文档事务操作，弱一致性	支持事务操作
占用空间	占用空间大	占用空间小
join操作	MongoDB没有join	MySQL支持join

通过 Docker 启动 MongoDB

- 官方指引
  - https://hub.docker.com/_/mongo
- 获取镜像
  - docker pull mongo
- 运行MongoDB镜像
  - docker run --name mongo -p 27017:27017 -e MONGO_INITDB_ROOT_USERNAME=admin -e MONGO_INITDB_ROOT_PASSWORD=admin -d mongo

通过 Docker 启动 MongoDB

登录到 MongoDB 容器中

- `docker exec -it mongo bash`

通过 Shell 连接 MongoDB

- `mongo -u admin -p admin`

Spring 对 MongoDB 的支持

- MongoDB 是一一款开源的文文档型数据库
  - https://www.mongodb.com
- Spring 对 MongoDB 的支持
  - **Spring Data MongoDB（Starter）**
  - **MongoTemplate**
  - **Repository 支持**

Spring Data MongoDB 的基本用法

- 注解
  - @Document
  - @Id
- MongoTemplate
  - save / remove
  - Criteria / Query / Update

初始化 MongoDB 的库及权限

mongo-demo

- 创建库

```
use springbucks;
```

- 创建用户

```
db.createUser(
    {
      user: "springbucks",
      pwd: "springbucks",
      roles: [
        { role: "readWrite", db: "springbucks" }
      ]
} )
```

Spring Data MongoDB 的 Repository

mongo-repository-demo

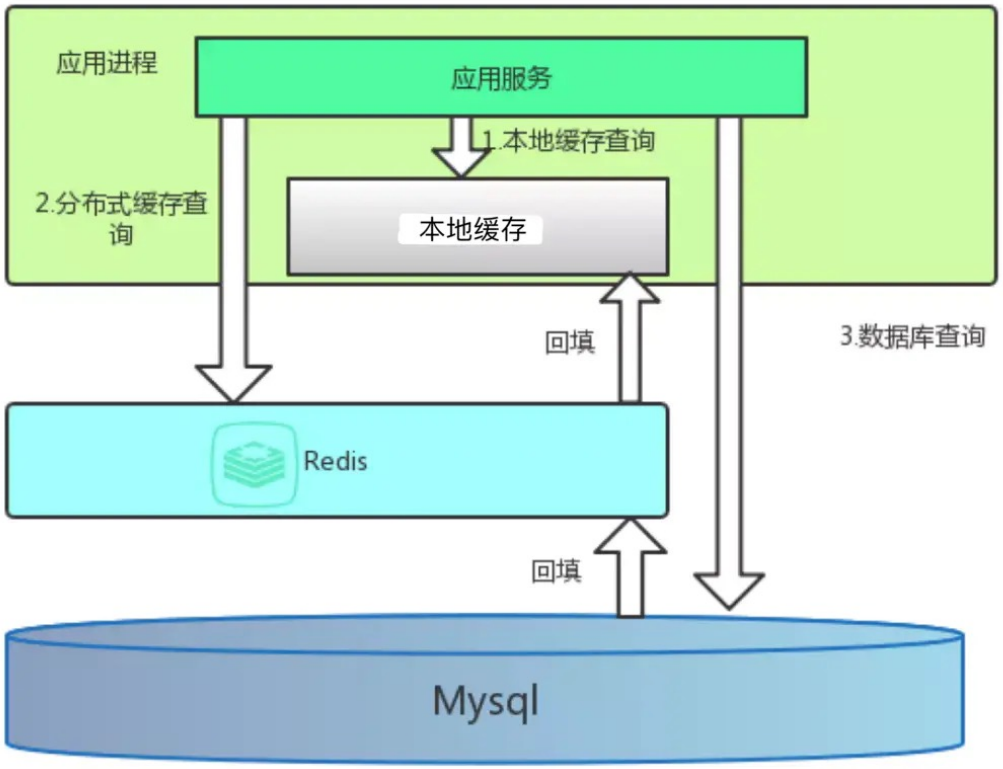
- 注解
  - @EnableMongoRepositories
- 对应接口
  - MongoRepository<T, ID>
  - PagingAndSortingRepository<T, ID>
  - CrudRepository<T, ID>

内容提要

- 在 Spring 中访问 MongoDB
- 在 Spring 中访问 Redis
- Spring 的缓存抽象
- 基于Redis的缓存

非关系型数据库 - Redis

- **缓存数据库**
  - 缓存就是数据交换的缓冲区(cache)当浏览器执行请求时,首先会对在缓存中进行查找,如果存在就获取;否则就访问数据库。
  - 缓存的好处就是读取速度快。



通过 Docker 启动 Redis

- 官方指引
  - https://hub.docker.com/_/redis
- 获取镜像
  - docker pull redis
- 运行镜像
  - docker run --name redis -d -p 6379:6379 redis

Spring 对 Redis 的支持

- Redis 是一一款开源的内存 KV 存储，支支持多种数据结构
  - https://redis.io

- Spring 对 Redis 的支支持
  - **Spring Data Redis（Starter）**
  - 支持的客户端 **Jedis / Lettuce**
  - **RedisTemplate**
  - **Repository** 支持

内容提要

- 在 Spring 中访问 MongoDB
- 在 Spring 中访问 Redis
- Spring 的缓存抽象
- 基于Redis的缓存

Spring 的缓存抽象

- 为不同的缓存提供一层抽象
  - 为 Java 方法增加缓存，缓存执行结果
  - 支持ConcurrentMap、EhCache、Caffeine、JCache(JSR-107)

- 接口
  - org.springframework.cache.Cache
  - org.springframework.cache.CacheManager

基于注解的缓存

cache-demo

- @EnableCaching
  - **@Cacheable**
  - **@CacheEvict**
  - **@CachePut**
  - **@Caching**
  - **@CacheConfig**

内容提要

- 在 Spring 中访问 MongoDB
- 在 Spring 中访问 Redis
- Spring 的缓存抽象
- 基于Redis的缓存

通过 Spring Boot 配置 Redis 缓存

```
<dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-cache</artifactId>
</dependency>
<dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-data-redis</artifactId>
</dependency>
```

```
spring.cache.type=redis
spring.cache.cache-names=coffee
spring.cache.redis.time-to-live=5000
spring.cache.redis.cache-null-values=false

spring.redis.host=localhost
```

通过 Spring Boot 配置 Redis 缓存

cache-redis-demo

```
@Slf4j
@Service
@CacheConfig(cacheNames = "coffee")
public class CoffeeService {
    @Autowired
    private CoffeeRepository coffeeRepository;

    @Cacheable
    public List<Coffee> findAllCoffee() {
        return coffeeRepository.findAll();
    }

    @CacheEvict
    public void reloadCoffee() {
    }
```

RedisTemplate

redis-demo

RedisTemplate<K, V>

- opsForXxx()

StringRedisTemplate

一定注意设置过期时间！！！

Redis Repository

redis-repository-demo

实体注解

- @RedisHash
- @Id
- @Indexed