How to Use Redis in Ruby On Rails Application

Redis is an open source very fast, non-relational, in-memory data structure store. It stores data in the form of a key-value pair. Redis has a support of data structure like sets, hashes, strings, list. Redis is mostly used for the cache store.

When to use Redis?

Redis is used when we have a performance issue related to scalability. For example when we have data in the database which is not going to change frequently then you use Redis to cache that data to remove the load from the MySql or PgSql server.

How to use Redis in Ruby on rails?

In order to use Redis in Ruby on Rails application you have to follow the steps given below:

Step 1:

Add the following gem into the **Gemfile** of application and run the "**bundle install**" command.

Step 2:

Next is to create **redis.rb** file in **config/initializers** directory and add the following code into that file.

In the above code "**app_name**" is the Namespace of my application. Gem "**redis-namespace**" organize everything under one application wise namespace, when multiple applications use same Redis server.

Step 3:

Next, to instruct rails to use Redis as a cache store we have added following code into **application.rb** file.

```
require_relative 'boot'

module Vexpo
  class Application < Rails::Application
    config.cache_store = :redis_store, 'redis://localhost:6379/0/cache', { expires_in: 90.minutes }
  end
end
```

Step 4:

In the below code I have written a query in **booths_controller.rb** file to fetch all booths of size 16m2. And write a helper method in **booths_helper.rb** to fetch the data from Redis.

```
class BoothsController < ApplicationController

  include ApplicationHelper
  include BoothsHelper

  def new_booth

    @booth_size_16 = fetch_booths_16

  end

end
```

```
module BoothsHelper

  def fetch_booths_16
    booth_16 = $redis.get("booth_16") rescue nil
    if booth_16.nil?
      booth_16_size = StaticBooth.where(:size_id => size_id(16)).where.not("booth_name like ?", "%W%")
      booth_16 = booth_16_size.to_json
      $redis.set("booth_16", booth_16)
      $redis.expire("booth_16",3.hour.to_i)
    end
    @booth_16 = JSON.load booth_16
  end

end
```

In the above code when we call the **new_booth** method of **booths_controller**, it will call the **fetch_booths_16** helper method.

In this **fetch_booths_16** method, the **booths_16** will get nil value because no data is pushed into redis yet. We then instruct rails to push **booths_16** data into redis. Then after a subsequent call to this method, data will be fetched from redis.

Since we are loading data in **new_booths** as a json we have to change when you call **@booth_size_16.name** to **@booth_size_16[“name”]** because data is in the form of json.

Advantages of using Redis:

- Support for rich data structure types: hash, string, set, list, sorted etc.
- Data is stored in memory and retrieve quickly: Redis store data in memory so instead of requesting data from the server it will fetch from memory and retrieve data very quickly
- Redis is used to speed up service response.
- Redis is a very fast non-relational and NoSQL key-value pair data store.
- Redis allows to insert a large amount of data into its cache very easily: Sometimes a situation comes like we have to load a thousand pieces of data into cache in a short period of time. This can be done by Redis, with mass insertion feature of it.
- Redis uses “Redis hashing” which is its own hashing mechanism: Datastore in redis in the form of key-value pair i.e. string field and a string value.

Redis offers efficient caching mechanism and it will take very less time to implement this cache mechanism. But outcomes of using this will get high-performing cache system in our application.

So this is the way you can improve the performance and increase the scalability of Rails application to a great extent.