Alex Bates

Databases II

8/30/2022

Redis Paper

Introduction/What Type

Redis is a key-value store database that is used for its very fast speeds that are due to the data being stored entirely in RAM. Redis stands for Remote Dictionary Server, this is because the entire server is just a massive dictionary. Part of the speed behind Redis is partially due to storing the data like this, and the other factor for Redis’s speed is the fact that the data is stored in ram, of course the data can, and many times is also stored on the hard disk for persistence and data durability.

Redis is really simple when it comes to performing CRUD operations, it uses 3 commands to perform those operations. We use the GET command to retrieve the data, DEL to delete data, and the SET command to create or update data. Do not worry, the SET command has options that can modify it’s behavior, the most important of those are the NX and XX options. NX makes sure the SET command can only create new items, it will fail if an item with the key already exists. XX does the opposite, the SET command will now only update where there are already records, it will not create any new ones. This simplicity is really cool, but it means that we have to do a lot of the heavy lifting with our code, bulk importing data is possible in a handful of ways, the most common being the Redis Input Output Tool (RIOT) or programmatically with your application.

Redis is almost entirely used for database caching due to it being so fast and weird to query on. While relational databases have rows and columns and relationships binding all your data together Redis has a big key-value map and stores strings, strings, and more strings. With the proper formatting and index building you can start creating complex data structures. With a series of commands and an index we can create complex queries and query aggregates. First, we create an index that sorts our data on something relevant to the group we want to make, then we call our aggregate function and give it our index and whatever part of the index we would like to group and sort by.

The pros of Redis are its speed and light overhead, its cons are that it is not very good for persistent storage. Nearly every database from MongoDB to MySQL to Neo4J automate or simplify so much for you such as data organization, complex queries, indexes, and clustering. Take mongo for instance, it will be much easier to implement rich data features into your projects because you don’t have to build indexes one every piece of information you want to query on. Redis is great for temporary data, and it is used for precisely that. Twitter uses Redis to store user DMs, ad counts, and engagement/impression data on Redis servers. At the end of the day what makes Redis so widely used is the things I talked about in this paper, Redis is crazy fast, crazy temporary, and crazy good for anything that wants those two things.

Cat for viewing pleasure:

                \_                          
                \`\*-.                      
                 )  \_`-.                   
                .  : `. .                  
                : \_   '  \                 
                ; \*` \_.   `\*-.\_            
                `-.-'          `-.         
                  ;       `       `.       
                  :.       .        \      
                  . \  .   :   .-'   .     
                  '  `+.;  ;  '      :     
                  :  '  |    ;       ;-.   
                  ; '   : :`-:     \_.`\* ;  
       [Redis] .\*' /  .\*' ; .\*`- +'  `\*'   
               `\*-\*   `\*-\*  `\*-\*'