Memcached versus Redis

Let's compare Memcached and Redis.

We'll cover the following

Introduction

This lesson will discuss some of the widely adopted real-world implementations of a distributed cache. Our focus will be on two well-known open-source frameworks: Memcached and Redis. They're highly scalable, highly performant, and robust caching tools. Both of these techniques follow the client-server model and achieve a latency of sub-millisecond. Let's discuss each one of them and then compare their usefulness.

Memcached

Memcached was introduced in 2003. It's a key-value store distributed cache designed to store objects very fast. Memcached stores data in the form of a key-value pair. Both the key and the value are strings. This means that any data that has been stored will have to be serialized. So, Memcached doesn't support and can't manipulate different data structures.

Memcached has a client and server component, each of which is necessary to run the system. The system is designed in a way that half the logic is encompassed in the server, whereas the other half is in the client. However, each server follows the **shared-nothing architecture**. In this architecture, servers are unaware of each other, and there's no synchronization, data sharing, and communication between the servers.

Due to the disconnected design, Memcached is able to achieve almost a deterministic query speed $(O(1))$ serving millions of keys per second using a high-end system. Therefore, Memcached offers a high throughput and low latency.