Do I Need A Cache?

In this lesson, we will discuss how to tell if we need caching in our application.

We'll cover the following

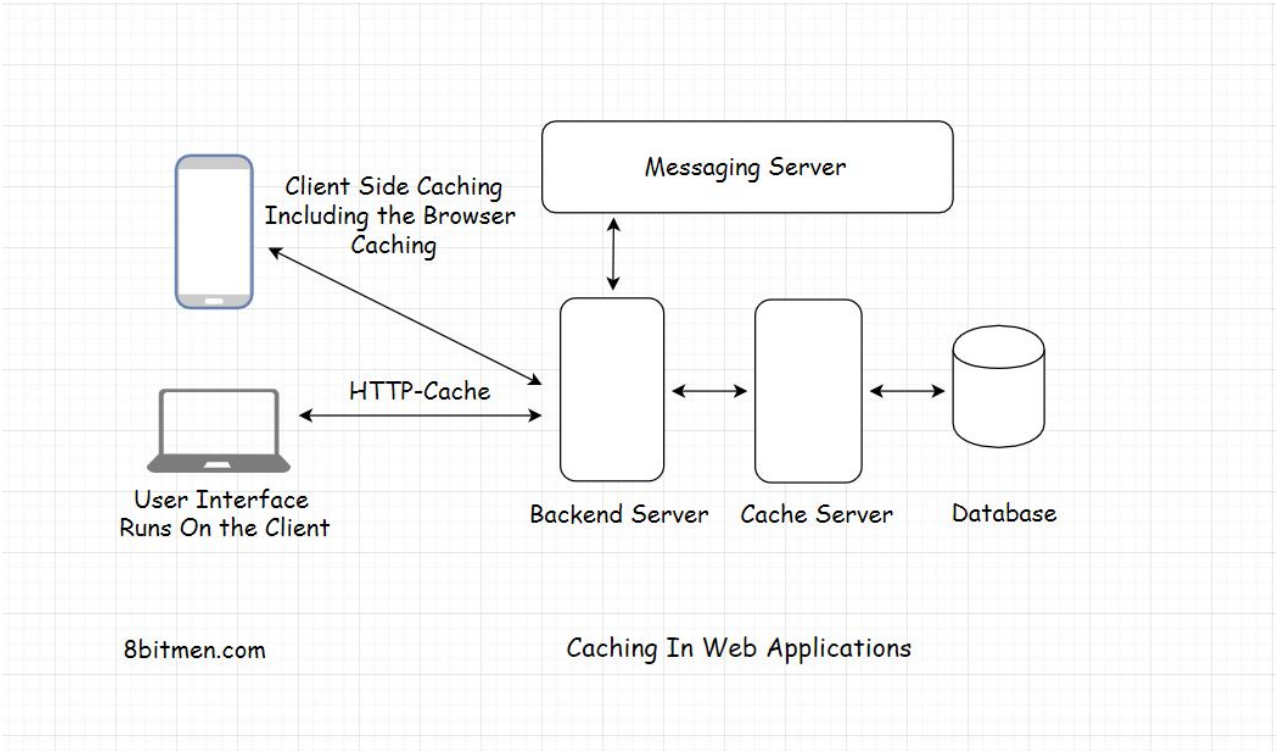
- Different components in the application architecture where the cache can be used

First up, it’s always a good idea to use a cache as opposed to not using it. It doesn’t do any harm. It can be used at any layer of the application, and there are no ground rules as to where it can and cannot be applied.

The most common usage of caching is database caching. Caching helps alleviate the stress on the database by intercepting the requests being routed to the database for data.

Then, the cache then returns all the frequently accessed data. Thus, cutting down the load on the database by notches.

Different components in the application architecture where the cache can be used#



Across the architecture of our application, we can use caching at multiple places. Caching is used in the client browser to cache static data. It is used with the database to intercept all the data requests, in the REST API implementation, etc.

Besides these places, I suggest you look for patterns. We can always cache the frequently accessed content on our website, be it from any component. There is no need to compute stuff over and over when it can be cached.

Think of *joins* in relational databases. They are notorious for making the response slow. More *joins* means more latency. A cache can avert the need for running *joins* every time just by storing the data in demand. Now, imagine how much would this mechanism speed up our application.

Also, even if the database goes down for a while, the users won't notice it as the cache would continue to serve the data requests.

Caching is also the core of the *HTTP* protocol. This is a good resource to read more about it. (https://developers.google.com/web/fundamentals/performance/optimizin

g-content-efficiency/http-caching)

We can store user sessions in a cache. It can be implemented at any layer of an application be it at the OS level, at the network level, CDN, or the database.

You might remember, we talked about *Key-value* data stores in the database lesson. They are primarily used to implement caching in web applications.

They can be used for *cross-module communication* in a *microservices* architecture by saving the shared data which is commonly accessed by all the services. It acts as a backbone for the *microservice* communication.

Key-value data stores via caching are also widely used in *in-memory data stream processing* and running *analytics*.

← Back

Introduction

Next →

Reducing the Application Deployment...

Mark as Completed

Report an Issue