

10/27/2021

Azamit below(112322)

## LAB-2

### HOW JS ENGINE OPTIMIZE OUR CODE.

#### What is JS engine

A JavaScript engine is a program or an interpreter which executes JavaScript code.

And the most recent and popularly used JavaScript engine these days is the V8 JavaScript engine which was initially developed for Google Chrome and Chromium web browsers to improve the performance of JavaScript execution. And it's written in C++.

What makes V8 different than the previous JavaScript engines is that it uses an interpreter and a compiler which optimizes our code to run faster. Every browser uses its own JS engine.

#### How it operates

Most JavaScript compilers perform the **just-in-time or (JIT)** compilation that happens at run time. So the great thing about JIT is, once the code starts running it can optimize it as it goes through the code.

**Interpreter** goes through the code line by line as it changes it to lower level language and this can be slower in performance.

So the **profiler** runs the code through interpreter and at run time keeps track of warm segments which runs a few times and hot segments which runs multiple times. So at this time JIT sends warm segments to **baseline compiler**. And hot segments to an **optimization compiler**.

#### How it optimizes the code

the crankshaft threat tries to optimize our code in one of these ways: -

**Inlining** is the first optimization, and it's the process of replacing the call site with the body of the called function.

**Hidden classes**, a hidden class is created every time a new property is added to an object, and the old hidden class is updated with a transition path to the new hidden class. Hidden class transitions are important because they allow hidden classes to be shared among objects that are

created the same way. If two objects share a hidden class and the same property is added to both of them, transitions will ensure that both objects receive the same new hidden class and all the optimized code that comes with it. *So, it's always a good practice not to change the shape of our object when we write code so that our code can be optimizable.*

**Inline caching** is the compiler (V8) maintains a cache of the type of parameters passed and uses that information to predict or make assumptions about the type of parameters that will be passed in the future.

So this will help to skip the process of trying to figure out the type of each parameter that is passed and instead just use the information we already have of the type to be passed. And if an object is passed as parameter inline caching and hidden classes work together to optimize our code.

It can also process and deoptimization to make the opposite transformation and reverts back to the non-optimized code in case an assumption the engine made doesn't hold true anymore.

### **Garbage collection**

As an addition, It also handles garbage collection, unlike other languages JavaScript automatically removes data that is not used or has no reference pointer. And takes to the garbage collector and frees space in the heap as the heap memory is limited.

And finally compiles our code and change to machine language