Mahlet Honelign
Id: 112174

10/29/2021

JavaScript Engines

- JavaScript engines are what make JavaScript efficient. They handle the performance of JavaScript code that starts from the *source code* and go through certain transitions in the process of being converted to *bytecode*.

- The first transition that happens when a JavaScript code runs is the source code being parsed to an abstract syntax tree (AST). Then the AST passes though the interpreter to get converted to a bytecode. The bytecode and a profiling data will be sent to an optimizing compiler, which makes an optimized code out of it to convert it to bytecode. *Optimized code* is simply a code that is memorized by the compiler as a result of repetitive tasks or functions (such functions are usually referred as "hot" functions). To avoid further repetitions, the compiler refers to this memorized information while the "hotness" of the function is still on. This process of optimization is what results in a better performance. If at any point, it reaches to a different function or task that does not match with the optimized code, *deoptimization* takes place, which means the path reverts to the point where it has to make another optimizing path for the new function.

- JavaScript engines optimize objects by using their *shape* (also called *structure*, *hidden class, type* etc.). When multiple objects have the same property keys, it can be said that they have the same shape. This shape is stored outside the JavaScript object, containing all the property attributes except the value. The value is stored in the object itself. In addition to the other property attributes, the shape contains what is called *Offset* (like an index, it starts from 0), which tells the position of the values of the property on the object. So, any object that has the same shape, will only have to store its values, and will use the shape that is already created for the property attributes. This avoids duplication and unnecessary memory usage.

- JavaScript engines vary from browser to browser, for example, V8 is used by Chrome, SpiderMonkey is used by FireFox, JavaScriptCore (JSC) is used by Safari… and so on.

- The architecture of all JavaScript engines is similar since they all have parser, interpreter, and compiler. Nevertheless, there is a slight difference regarding how many optimizing compilers they use. For example, V8 uses one optimizing compiler TurboFan, and SpiderMonkey uses a baseline compiler for a somewhat optimized code, and IonMonkey which produces a heavily optimized code.