How JS Engine Optimize JavaScript code

A **JavaScript engine** is a program or an interpreter which executes JavaScript code. A JavaScript engine can be implemented as a standard interpreter, or just-in-time compiler that compiles JavaScript to bytecode in some form.

From what I read I understand the process how Js engine optimize JavaScript code. And the process starts with Js file, and it fed to Parser and the parser tokenizes the code and changes it Abstract syntax tree(AST) and after that the Interpreter (called ignition in google chrome) and is responsible for generating and executing bytecode. which can be used to speed up the execution later. When a function becomes *hot* (JIT profiler is responsible for making the statement hot and warm) for example when it's run often, the generated bytecode **and** the profiling data are passed on to Optimizing compiler (which is called Turbofan), our optimizing compiler, to generate highly optimized machine code based on the profiling data. The Baseline compiler starts generating Stub, if the function is polymorphic, it'll create Stub for every possible combination.

Some tips for improving performance optimizations in v8

- Tip1: Declare classes in script scope
- Tip2: Keep object property ordering constant
- Tip3: Declare object properties in constructor
- Tip4: Fix function argument types

Why do some engines have more optimizing compilers than others? It's all about trade-offs. An interpreter can produce bytecode quickly, but bytecode is generally not very efficient. An optimizing compiler on the other hand takes a little longer, but eventually produces much more efficient machine code. There is a trade-off between quickly getting code to run (interpreter) or taking some more time, but eventually running the code with optimal performance (optimizing compiler). Some engines choose to add multiple

optimizing compilers with different time/efficiency characteristics, allowing for more fine-grained control over these trade-offs at the cost of additional complexity. But besides these differences, at a high level, **all JavaScript engines have the same architecture**: