Node Js (helps us write JS and make it work, gives us interface, interpret the code)

50% JS

50% C++

100% JS

JS code we write

Libuv (gives node the ability to handle OS files, networking)

30% JS

70% C++

V8 (open source engine to run JS outside the browser i.e. using CMD developed by google)

100% C++

* Node also gives us wrapper (standard library module) like http, fs, crypto, path which are consistent API and refer to functionality implemented in V8 or libuv, so we don’t have to directly go and implement C++ code, we write JS

Lecture 3

URL - <https://github.com/nodejs/node>

lib – contains JS definition of modules (http, fs, crypto, path), mostly js code, JS side

src – is c++ implementations of modules (http, fs, crypto, path), mostly c++ code, C++ side

we looked at pbkdf2 function

lib -> internal -> crypto -> pbkdf2.js

where the function pbkdf2 is exported

when we call this function node executes,

pbkdf2(password, salt, iterations, keylen, digest, callback) which has error checking

then

function \_pbkdf2(password, salt, iterations, keylen, digest, callback) again more error checking

then

const ret = PBKDF2(password, salt, iterations, keylen, digest, callback);

which comes from

process.binding('crypto'); this way node binds to C++

JS code we write

Connects (bridge between) JS and c++

Converts values between JS & C++ world

libuv

Node C++ side (src folder in node repo)

V8

Node JS side (lib folder in Node repo)

Node library JS side

Process.binding()

Node js library C++ side

lecture 4

process.binding('crypto'); this way node binds to C++

this acts as bridge between our JS and C++

we can find the c++ implementation of pbkdf2 in src->node\_crypto.cc

find void PBKDF2(const FunctionCallbackInfo<Value>& args) {

which has implementation of pbkdf2 in C++

if u scroll up we see using v8::Array;

which is using v8 engine

this file also uses libuv (quite difficult to find directly, but something like uv\_thread\_t is used throughout)

**Lecture 5**

Instance of running program

Process

thread

Multiply 2 by 2

Store result in variable

Divide variable by 2

Is variable equal to 2?

When ever we start a program In our OS, the OS start a process which is instance of the running program that is being executed it has multiple threads, the thread is given to CPU, and the CPU executes instructions in the thread.

Scheduling is used to determine which thread will be processed first

Example -

Process 2

Thread to perform action on keyboard keys

OS Scheduler decides which thread can wait or which one wont

Thread to move cursor when mouse moves

Process 1

So the when mouse moves (urgent thread) the priority for this is highest or else the user will feel the computer is broken.

CPU core helps more threads,

Second is examine the instructions (or work done) in each thread,

IO takes Time

Thread 1

Thread 2

Read file form HDD

Count number of letters

Multiply 3x3

So the OS scheduler knows the IO operations takes time greater than 0, so cpu will have to wait, so the scheduler executes thread 2 which wouldn’t make Cpu to wait, and then the cpu can resume the wait for the IO operations.

**Lecture 6**

Node Program

Event Loop

One Thread

Inside a single thread there is a Event loop, its like a control structure that decides what the thread should at any of point of given time. So every code we write has 1 event loop at its core.

entire body of code executes in one 'tick'

see event\_loop\_psuedo\_code.js for event loop description.

**lecture 9**

Node event loop

Some of node framework/standard libraries

Single threaded

Not single threaded

If node were single threaded refer thread\_calls.js

Thread #1

1. crypto.pbkdf2()
2. crypto.pbkdf2()

1s

1s

2s

Actually node performance was like this

Pbkdf2 #1

Pbkdf2 #2

1s

0s

Here is why this happens?

The libuv uses thread pool

Beside 1 thread to event loop, there are 4 more threads which are used for computational intensive tasks by some functions

So if we

JavaScript code we write

Pbkdf2

V8

Node C++ side

Thread pool

T1

T2

T3

T4

libuv

Node crypto module

Core 1

Core 2

Os thread scheduler

Thread pool

T1

T2

T3

T4

Pbkdf2 - 1

Pbkdf2 - 2

Pbkdf2 - 3

Pbkdf2 - 4

Pbkdf2 - 5

This 1 doesn’t get thread so executed late

Common thread pool questions

How does event loop relate to thread pool?

Event loop checks if any pending operations are left, that is any function is using thread pool