NodeJS

C/C++ Modules

Chin Ho Cheung
September 11th 2012

What's the difference ?

Practically, nothing changed:

- Same way to load
- Same way to use
- Same way to publish

Why C/C++?

- NodeJS based on V8 engine written in C++
- Use existing C/C++ libraries
- In some ways, using C/C++ is faster

What can it do ?

- Makes C/C++ functions accessible to javascript
- Pass arguments between JS and C/C++ worlds
- Callback functions called directly from C/C++

How do I obtain a .node from C/C++ ?

Simple as a compilation:

```
Compile ...

g++ -g -fPIC -DPIC -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64
-D_GNU_SOURCE -DEV_MULTIPLICITY=0 -I/usr/local/include/node
<srcName>.cc -c -o <srcName>.o

... and link:

g++ <srcName>.o -o <srcName>.node -shared -L/usr/local/lib
```

How do I obtain a .node from C/C++ ?

... actually it's easier:

Dedicated set of tools to compile from C/C++ to node:
- node-waf using waf building system(old)
- node-gyp using gyp

You JUST have to create some configuration files ...

A (not very original) example

Getting a random number from C/C++ module ...

A (not very original) example

Some includes ...

```
1 #include <v8.h>
2 #include <node.h>
3 #include <stdlib.h>
4
```

A (not very original) example

Letting node know that my module can be used:

```
static void
Init( v8::Handle<v8::Object> target ) {
    NODE_SET_METHOD(target, "randomNumber", RandomNumber);
}

NODE_MODULE(simpleModule, Init);
```

A (not very original) example

At last, the random:

```
static v8::Handle<v8::Value>
RandomNumber( const v8::Arguments& args ) {
    // Everything inside RandomNumber is within this scope
    // The intelligence behind it is that eveything
    // will be cleared once method exits
    v8::HandleScope scope;

    // Method Close() of HandleScope returns the value in parameter
    // inside the previous scope (the caller's)
    return scope.Close( v8::Integer::New( rand() % 10 ) );
}
```

A (not very original) example

The "famous" node-gyp configuration file

```
{
  "targets": [
    {
      "target_name": "simpleModule",
      "sources": [ "src/simple.cpp" ]
    }
  ]
}
```

Just for the fun, another one

Passing arguments from JS to C/C++:

```
using namespace v8;

static Handle<Value>
TransformString( const Arguments& args ) {
    HandleScope scope;

    // args[0] is a v8::Value with to* methods
    Local<String> str = args[0]->ToString();

    return scope.Close( String::Concat( String::New("Yep, I have modified: "), str ));
}
```

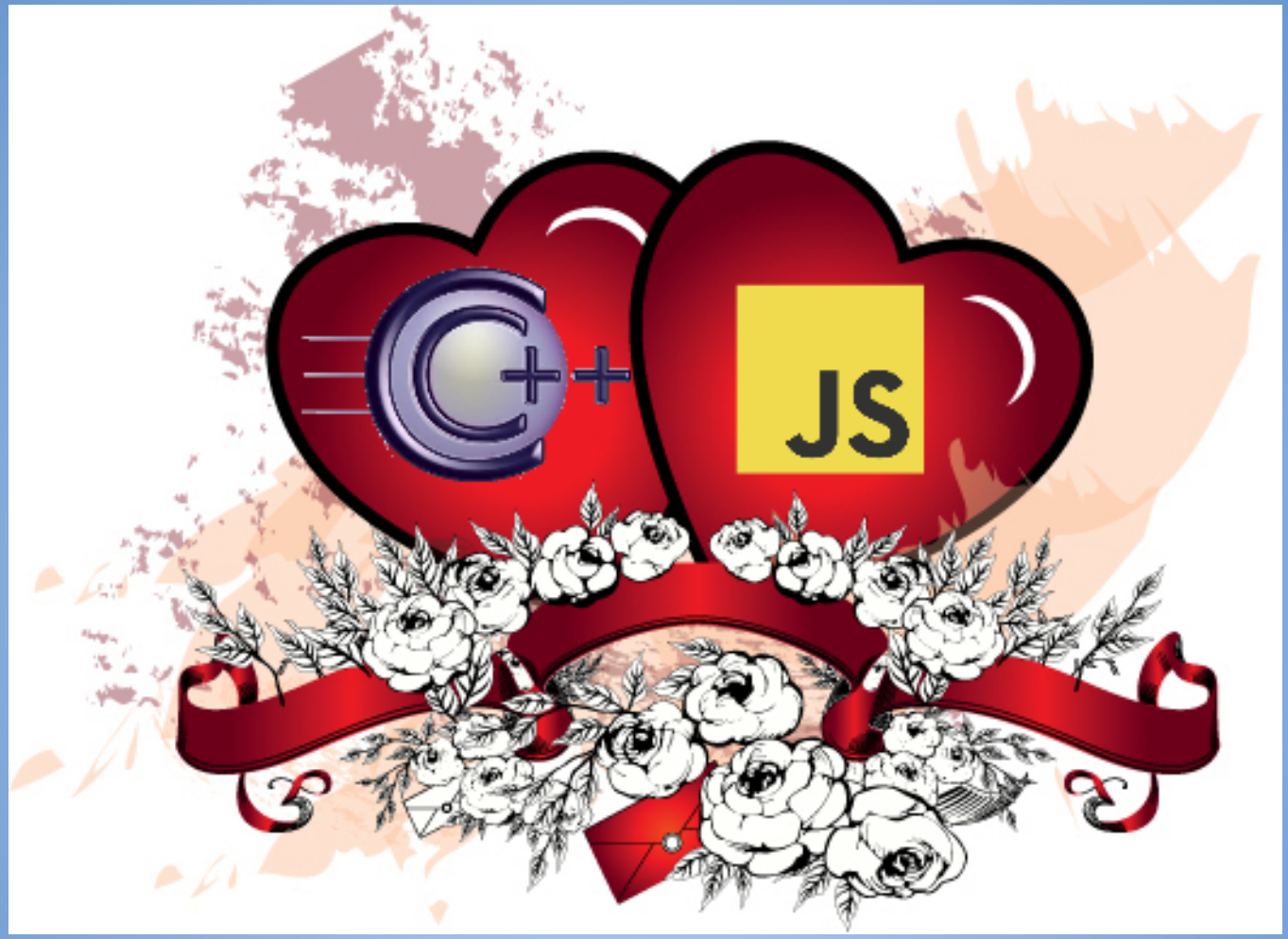
What haven't I talked about ?

- In depth description of what is a v8::Handle
- Making my C/C++ code JS-friendly using v8::context
- Do complicated stuff with libuv

And other things I've forgotten ...

Resources

http://nodejs.org/
https://developers.google.com/v8/
http://izs.me/v8-docs/main.html
https://github.com/chinhoc/pres-node-cpp-modules



Questions ?