

Plugins - C/C++, Lua, V8, wasm

Kit Chan (kichan@apache.org)

Current State of the Plugin World

- Plugins to remove or promote (Wednesday)
 - Remap config changes (Wednesday)
 - Plugin reload (Wednesday)
-
- C Plugins
 - C++ Plugins
 - Lua Plugin
 - Question 1: Do we want a repository of Lua scripts to share?
 - Question 2: Need some Improvements?

Repository for Lua scripts

- Starting point? - https://github.com/shukitchan/ats_lua_scripts
 - Connecting to 3rd libraries like GeoIP, redis, torch
 - Uncompressing response in transform plugin
 - Demonstrate how to do stale-while-revalidate and stale-if-error
 - Etc
- New Location?
- Any one else want to contribute? Guidelines on what stuff to put in?

Lua Plugin improvements

- Bad naming - “uri” used as the actual url path e.g. `ts.remap.get_to_uri()`
- Lack of Encapsulation
 - Request
 - E.g. `local req = ts.client_request; local m = req:method(); req:method('GET')`
 - Url
 - E.g. `local url = ts.client_request.url; local path = url:path(); url:path('/index.html')`
 - Addr
 - E.g. `local addr = ts.server_request.server_addr; local ip = addr:ip();
addr:ip("10.10.10.10")`
 - Response
 - E.g. `local resp = ts.server_response; local status = resp:status(); resp:status('400')`
 - Config
 - E.g. `local value = ts.config['TS_LUA_CONFIG_HTTP_CACHE_HTTP']`
 - E.g. `ts.config['TS_LUA_CONFIG_HTTP_CACHE_HTTP'] = 0`
- Anything Else ?

Lua Plugin Limitation

- 2GB limit in memory for luajit
- Need support for other popular languages
- Lack of development in luajit (Last release in 2017-05)
- Relative lack of popularity on the entire ecosystem as a whole

```
1 -- Example showing how to use torch inside ATS Lua script. Depends on torch
2
3 -- Setup instructions
4 -- 1) Install torch following instructions here - http://torch.ch/docs/getting-started.html
5 -- (in the example below, I installed torch under /home/root/)
6 -- 2) Install nn - sudo luarocks install nn
7
8 ts.add_package_path('/home/root/.luarocks/share/lua/5.1/?lua;/home/root/.luarocks/share/lua/5.1/?/init.lua;/home/root/torch/in
9 ts.add_package_cpath('/home/root/.luarocks/lib/lua/5.1/?so;/home/root/torch/install/lib/lua/5.1/?so;./?.so;/usr/local/lib/lua
10
11 require 'nn'
12 -- set up a model mentioned in http://mdtux89.github.io/2015/12/11/torch-tutorial.html
13 mlp = nn.Sequential()
14
15 inputSize = 10
16 hiddenLayer1Size = opt.units
17 hiddenLayer2Size = opt.units
18
19 mlp:add(nn.Linear(inputSize, hiddenLayer1Size))
20 mlp:add(nn.Tanh())
21 mlp:add(nn.Linear(hiddenLayer1Size, hiddenLayer2Size))
22 mlp:add(nn.Tanh())
23
24 nclasses = 2
25
26 mlp:add(nn.Linear(hiddenLayer2Size, nclasses))
27 mlp:add(nn.LogSoftMax())
28
29 function do_global_read_request()
30     -- use the model
31     out = mlp:forward(torch.randn(1,10))
32     ts.debug(out[1][1])
33 end
```

New Idea - WebAssembly: Hello World

```
<html><head>
<script>
var buffer = new
Uint8Array([0,97,115,109,1,0,0,0,1,133,128,128,128,0,1,96,0,1,127,3,130,128,128,128,0,1,0,4,132,128,128,128,0,1,112,0,0,5,131,128,128,128,0,1,
0,1,6,129,128,128,128,0,0,7,146,128,128,128,0,2,6,109,101,109,111,114,121,2,0,5,104,101,108,108,111,0,0,10,138,128,128,128,0,1,132,128,128,
128,0,0,65,16,11,11,146,128,128,128,0,1,0,65,16,11,12,72,101,108,108,111,32,87,111,114,108,100,0]);
var m = new WebAssembly.Instance(new WebAssembly.Module(buffer));
var h = new Uint8Array(m.exports.memory.buffer);
var p = m.exports.hello();
function utf8ToString(h, p) {
  let s = "";
  for (i = p; h[i]; i++) {
    s += String.fromCharCode(h[i]);
  }
  return s;
}
console.log(utf8ToString(h, p));
</script>
</head><body>This is a test</body></html>
```

wasm2wabt

```
(module
  (type $type0 (func (result i32)))
  (table 0 anyfunc)
  (memory 1)
  (export "memory" memory)
  (export "hello" $func0)
  (func $func0 (result i32)
    i32.const 16
  )
  (data (i32.const 16)
    "Hello World\00"
  )
)
```

emcc

```
char * hello() {
  return "Hello World";
}
```

WebAssembly for CDN

- Edge Computing: Programming on the edge is a thing
- Isolation important for CDN with many tenants - i.e. Scale
- VM/Container too slow / heavy for this case
- People likes to program in different popular languages

Reference

- <https://thestrangeloop.com/2018/isolation-without-containers.html>
- <https://hacks.mozilla.org/2018/10/webassemblies-post-mvp-future/>

Stackpath EdgeEngine & Cloudflare Worker

- Based on V8
- Liftoff Engine for WebAssembly
- Demo Walkthrough (C example) - <https://github.com/cloudflare/cloudflare-workers-wasm-demo>
- Cloudflare Wrangler - CLI to generate develop, test and deploy Rust programs for worker
 - Reference - <https://developers.cloudflare.com/workers/webassembly/tutorial/>

- Does Linkedin already have a javascript plugin?

Prototype - ATS V8 Plugin

https://github.com/shukitchan/ats_v8_plugin

Can I skip the Javascript part?

- WebAssembly C/C++ API - API for embedding a wasm engine into application
 - Reference - <https://github.com/WebAssembly/wasm-c-api/blob/master/example/hello.c>
 - Works with V8 now
 - Will support other wasm runtime eventually

Fastly Lucet / Terrarium

- Lucet - tool chains to convert wasm to native code and runtime for that
- Terrarium - Provide environment for Edge programming using Lucet
- Instantiation of WebAssembly module is faster than V8 but ...
- Supports WASI
- Based on Cranelift
- Not much documentation on how to embed this to application yet
- Reference
 - <https://github.com/fastly/lucet>
 - <https://wasm.fastlylabs.com>

Other WebAssembly Runtime

- Wasmjit (Can't find the repo any more)
- Wasmi (wasm interpreter in Rust)
- VMIR (optional JIT)
- WAVM (Based on LLVM)
- Wasmtime (Uses Cranelift and supports WASI)
- Wasmer.io (Flexible to support different backend, Rust crates under the hood)

- (Envoy POC with WAVM - <https://github.com/envoyproxy/envoy/issues/4272>)

Envoy POC

https://github.com/PiotrSikora/envoy/tree/wasm-missing_import

Discussions

Other Topics in Plugins

- Plugin Coordination
- Scriptlet -

<https://cwiki.apache.org/confluence/display/TS/Presentations+-+2015?preview=/56066455/56131600/Lua%20Scriptlets.pdf>