WebAssembly for Proxy (or “proxy-wasm”)

Kit Chan (kichan@apache.org)

Spring ATS Summit, June 24 2021

Envoy Http Filter

- One of the ways to extend functionality of Envoy
  - E.g. add header to request/response, change destination of routing
  - Similar to ATS plugins
- C++
- Compiled and shipped as part of the binary
- Many prebuilt
  - List - https://www.envoyproxy.io/docs/envoy/latest/configuration/http/http_filters/http_filters
  - E.g. AWS Request Signing, OAuth2 , Rate limiting, etc
  - Lua
  - Wasm

Wasm Filter for Envoy

Example filter configuration:

wasm *envoy.yaml*

```
                cluster: web_service

        http_filters:
        - name: envoy.filters.http.wasm
          typed_config:
            "@type": type.googleapis.com/envoy.extensions.filters.http.wasm.v3.Wasm
            config:
              name: "my_plugin"
              root_id: "my_root_id"
              # if your wasm filter requires custom configuration you can add
              # as follows
              configuration:
                "@type": "type.googleapis.com/google.protobuf.StringValue"
                value: |
                  {}
              vm_config:
                runtime: "envoy.wasm.runtime.v8"
                vm_id: "my_vm_id"
                code:
                  local:
                    filename: "lib/envoy_filter_http_wasm_example.wasm"
        - name: envoy.filters.http.router
          typed_config: {}

clusters:
- name: web_service
```

Wasm Runtime

- For running WebAssembly code
- Some can compile into executable native machine code
- Choices:
  - V8 - Wasm VM from Chrome, loads fast
  - WAVM - pre-compiles wasm to native assembly, loads slow but presumably run faster
  - Others - wasmtime, wamr

WAVM

Writing Your Code

```
class ExampleRootContext : public RootContext {
public:
  explicit ExampleRootContext(uint32_t id, std::string_view root_id) : RootContext(id, root_id)

  bool onStart(size_t) override;
  bool onConfigure(size_t) override;
  void onTick() override;
};

class ExampleContext : public Context {
public:
  explicit ExampleContext(uint32_t id, RootContext *root) : Context(id, root) {}

  void onCreate() override;
  FilterHeadersStatus onRequestHeaders(uint32_t headers, bool end_of_stream) override;
  FilterDataStatus onRequestBody(size_t body_buffer_length, bool end_of_stream) override;
  FilterHeadersStatus onResponseHeaders(uint32_t headers, bool end_of_stream) override;
  FilterDataStatus onResponseBody(size_t body_buffer_length, bool end_of_stream) override;
  void onDone() override;
  void onLog() override;
  void onDelete() override;
};
static RegisterContextFactory register_ExampleContext(CONTEXT_FACTORY(ExampleContext),
```

https://github.com/proxy-wasm/proxy-wasm-cpp-sdk/blob/master/example/http_wasm_example.cc

```
FilterHeadersStatus ExampleContext::onResponseHeaders(uint32_t, bool) {
  LOG_DEBUG(std::string("onResponseHeaders ") + std::to_string(id()));
  auto result = getResponseHeaderPairs();
  auto pairs = result->pairs();
  LOG_INFO(std::string("headers: ") + std::to_string(pairs.size()));
  for (auto &p : pairs) {
    LOG_INFO(std::string(p.first) + std::string(" -> ") + std::string(p.second));
  }
  addResponseHeader("X-Wasm-custom", "FOO");
  replaceResponseHeader("content-type", "text/plain; charset=utf-8");
  removeResponseHeader("content-length");
  return FilterHeadersStatus::Continue;
}
```

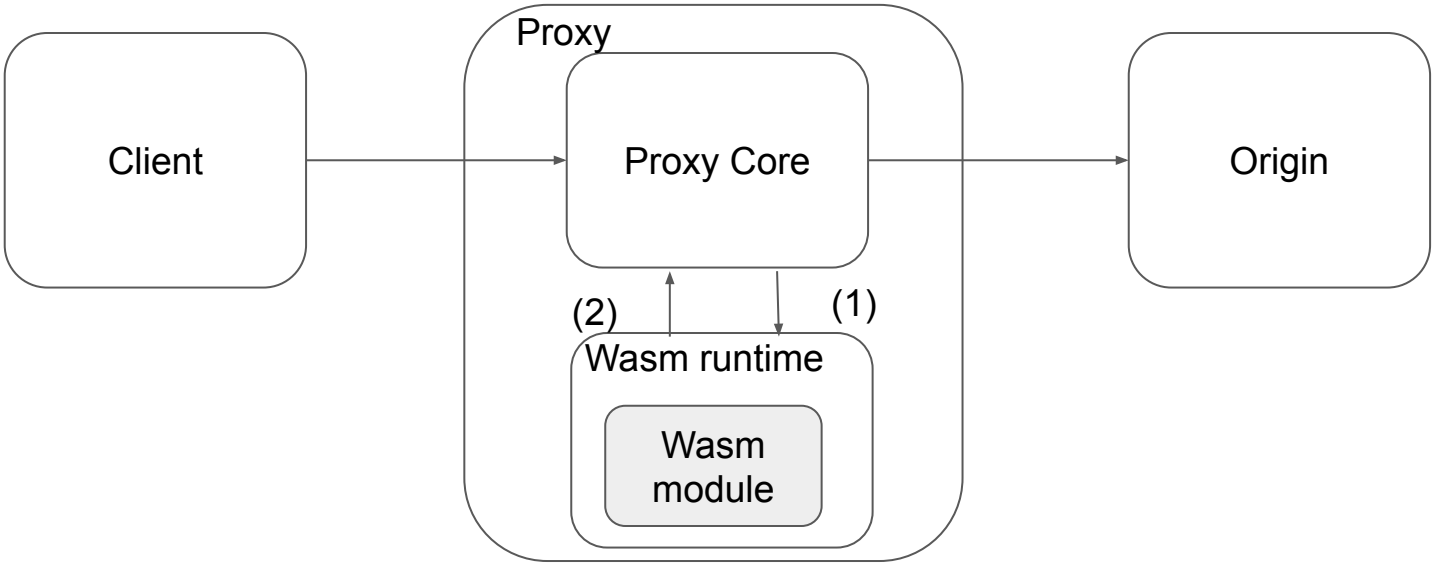
https://github.com/proxy-wasm/proxy-wasm-cpp-sdk/blob/master/example/http_wasm_example.cc

Proxy-wasm SDK

- To compile your code into “wasm” module
- Separate repositories on github
- Provide documentation for the development of WASM modules and APIs available
- Language supported
  - C++
  - Rust
  - Assemblyscript
  - GO
  - Zig
  - Check the list - https://github.com/proxy-wasm/spec/blob/master/README.md

Proxy-wasm Spec

- The SDK compiles your code into wasm modules
  - With specific functions for proxy to call (1)
  - Calling API that the proxy provides (2)
- Follow a Spec - sort of like “WASI” for proxy



Proxy-wasm Spec / ABI

- https://github.com/proxy-wasm/spec/tree/master/abi-versions/vNEXT
- Functions implemented by the module
  - E.g.
  - proxy_on_vm_start
  - proxy_on_done
  - proxy_on_http_request_headers
  - ...
- Functions implemented by proxy
  - E.g.
  - proxy_log
  - proxy_get_current_time
  - proxy_get_map
  - proxy_get_map_value
  - …

Proxy-wasm - Proxy

- We can follow the spec and implement proxy that supports proxy-wasm modules
- Some libraries are provided - https://github.com/proxy-wasm/proxy-wasm-cpp-host
- Implementations
  - Envoy / Istio Envoy
  - MOSN
  - ATS

ATS Wasm Plugin

- John Plevyak did that already (sort of)
  - https://github.com/jplevyak/trafficserver/tree/wasm/plugins/experimental/wasm
- Demo Time
- Dependencies
  - Plugin - WAVM / LLVM
  - SDK - emscripten

What’s Missing

- Catching up with the Spec
- Support more wasm runtimes - only WAVM supported for now
- Production testing
- Contributing to open source

Why Do It?

- Build plugins for ATS using other programming languages
- Wasm module can be “shared” between different proxies (Pipe Dream?)
- Support Advanced use case
  - E.g. doing inference on pre-built AI model

```
-- Example showing how to use torch inside ATS Lua script. Depends on torch

-- Setup instructions
-- 1) Install torch following instructions here - http://torch.ch/docs/getting-started.html
-- (in the example below, I installed torch under /home/root/)
-- 2) Install nn - sudo luarocks install nn

ts.add_package_path('/home/root/.luarocks/share/lua/5.1/?.lua;/home/root/.luarocks/share/lua/5.1/?/init.lua;/home/root/torch/install/share/lua/5.1/?.lua;/h
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/5.1/?.so;/usr/local/lib/lua

require 'nn'
-- set up a model mentioned in http://mdtux89.github.io/2015/12/11/torch-tutorial.html
mlp = nn.Sequential()

inputSize = 10
hiddenLayer1Size = opt.units
hiddenLayer2Size = opt.units

mlp:add(nn.Linear(inputSize, hiddenLayer1Size))
mlp:add(nn.Tanh())
mlp:add(nn.Linear(hiddenLayer1Size, hiddenLayer2Size))
mlp:add(nn.Tanh())

nclasses = 2

mlp:add(nn.Linear(hiddenLayer2Size, nclasses))
mlp:add(nn.LogSoftMax())

function do_global_read_request()
    -- use the model
    out = mlp:forward(torch.randn(1,10))
    ts.debug(out[1][1])
end
```

https://github.com/shukitchan/ats_lua_scripts/blob/master/torch_example.lua

Summary - Proxy-wasm

- The Spec
- The SDKs
- The proxy implementation

https://github.com/proxy-wasm