ZIA-PROJECT

TECHNICAL DOCUMENTATION

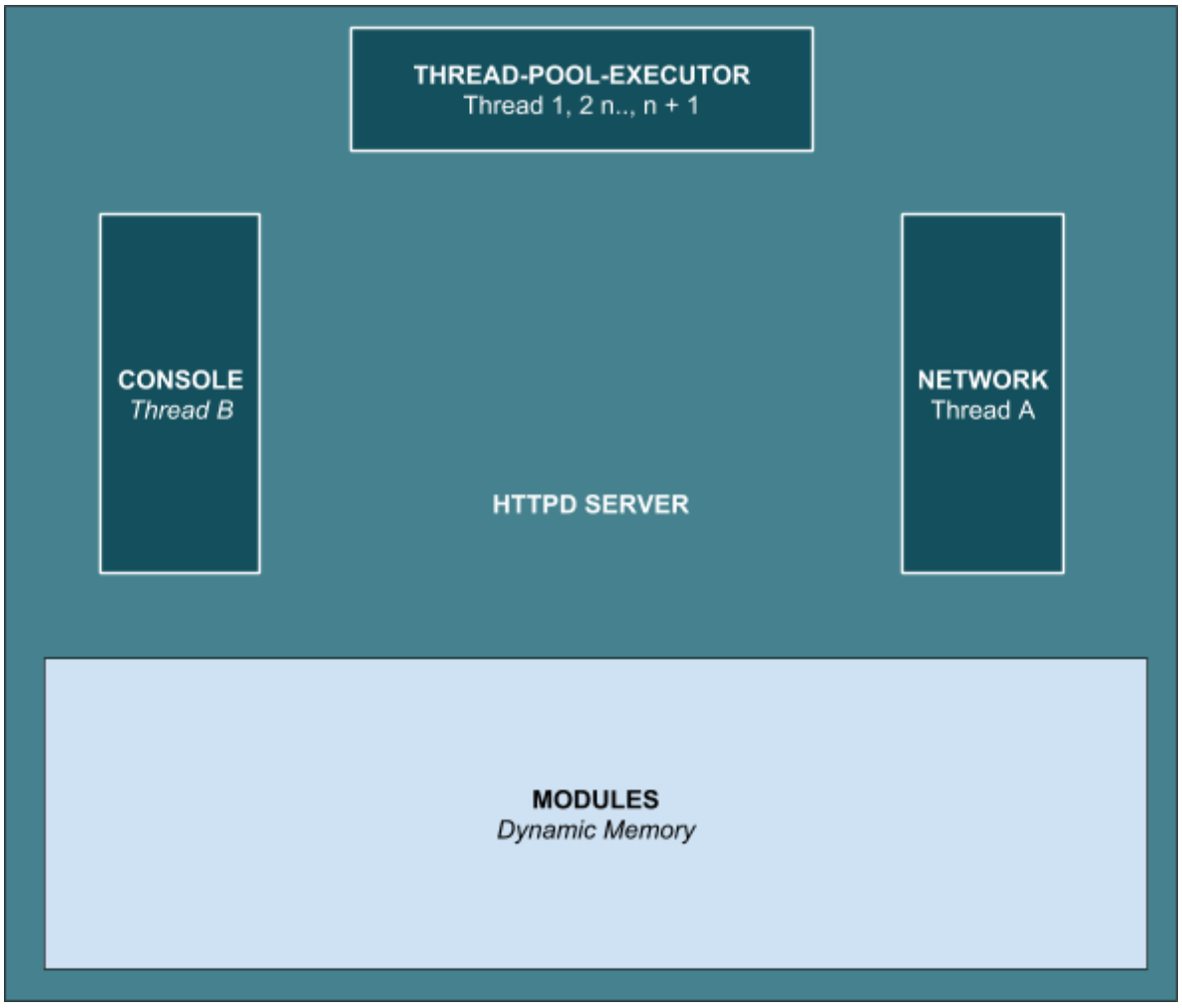
INTRODUCTION

This project is an httpd server following the http/1.1 rfc (https://tools.ietf.org/html/rfc2616). Similar applications already exist, such as Apache or Ngnix.

The goal of the zia is to communicate with web clients - e.g chrome, firefox - and serve web data content.

ARCHITECTURE

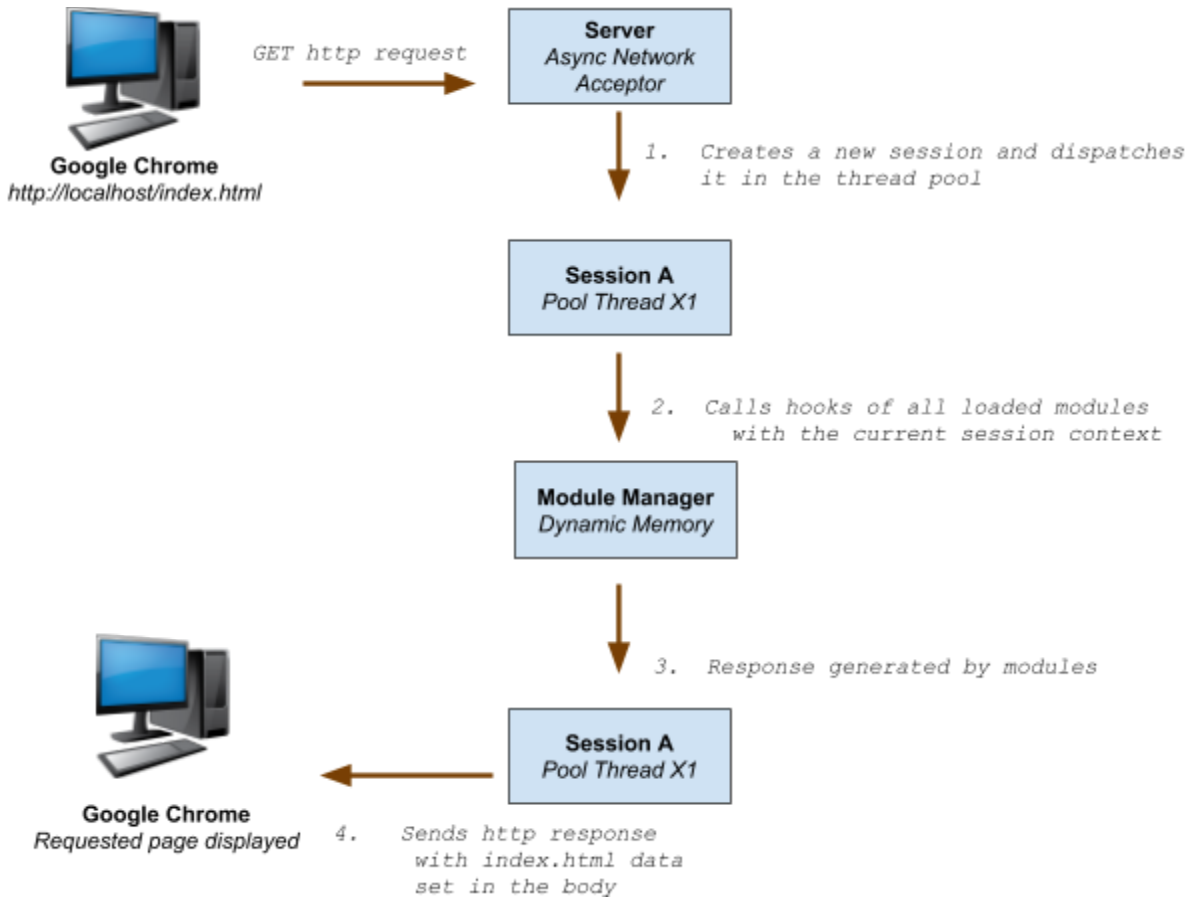
See below the global architecture and thread owners.



Quick explanation

- While running, the zia guarantees to have 1 thread alive, the console thread so it's always possible to type commands and execute a given list of actions.
- The network have 1 thread with the state alive or stopped. That's to say the network can refuse connections if the owner decides to stop it.
- When accepting a new network client, the network session process is dispatched unto a thread pool of X threads defined by the owner in the config of the server. So basically, a new session implies a new thread.
- Zia extensions can be added on linux and windows respectively with *.so and *.dll - This memory is owned by a module manager and is totally thread-safe. So it can be accessed by any session.

A request example



When the response is delivered, the session is automatically destroyed and the thread is free again for the next request. This mechanism is repeated as many times as the server is accepting connections.

CONFIGURATION

The server can be configured by editing the config.json file at the root of the project.

```
config.json

{
   "serverName": "MyHttpServer", -- Server Name
   "serverVersion": "1.1",       -- Server Version
   "address": "127.0.0.1",       -- Network bind address
   "port": 8080,                 -- Network bind port
   "poolSize": 10,               -- Thread pool executor size
   "networkReadBuffer": 8096,    -- Max bytes on each network read operation
   "modulesPath": "modules",     -- Relative path where the modules are located
   "modules": {                  -- Add here modules to load automatically
       "zia-log": 10,
       "zia-secure_network": 20,
       "PhpModule": 30,          -- The value corresponds to the priority
       "zia-unsecure_network": 40
   },
   "modulesProperties": {        -- Dynamic properties needed by the modules

       "tls_certificate_path": "tls/cert.crt",     -- path to the tls certificate
       "tls_private_key_path": "tls/private.key",  -- path to the tls private key
       "php_cgi_path": "/usr/bin/php-cgi",         -- path to php-cgi
       "php_www_path": "www"                       -- path to php web sources

   }
}
```

RUNTIME COMMANDS

Typing help on the console, the following output will be displayed:

```
help
[console]: command list
  config                    :           reloads the configuration file
  start                     :           starts the tcp server
  stop                      :           stops the tcp server
  restart                   :           restarts the tcp server
  load [module] [priority] :           loads a module
  unload [module]           :           unloads a module
  loadall                   :           loads all modules
  unloadall                 :           unloads all modules
  list                      :           list all loaded modules
  exit                      :           exits this program
```

As described above, the server can be configured dynamically. Load, unload a module and restart the network server is possible at the runtime. Reloading the config can also be done without reloading a module or even the network.

USAGE

To access your website - i.e located at /www - type the address in a navigator such as google chrome or firefox:

```
https://127.0.0.1:8080/
```

You can access it without a secure context by disabling the tls-module with the following command

```
unload zia-secure_network
```

So the website is now accessible though

```
http://127.0.0.1/8080/
```

CREDITS

Romain PILLOT, Raphaël GOULMOT, Antonin RAPINI, Thomas DEBRAND-PASSARD, Romain CHENG, Clément GOMIS, Raphaël LEGRAND