Implementation of Server Herd Proxy with Twisted Framework

Jingyu Liu
UCLA

Abstract

Twisted is an event-driven network framework in Python that supports a many network protocols. In this paper, Twisted is evaluated from aspects like API usability, implementation flexibility, documentation, and performance. This paper presents an implementation of server herd proxy. The evaluation is based on this project.

1. Introduction to Twisted API

Twisted's main package is twisted.internet. It provides a class called reactor. Reactor maintains the event loop for a twisted program. The event loop tracks the client requests and dispatches requests to event handlers. The event handlers in Twisted is called deferreds. They are handlers for asynchronous methods under cases like onSuccess, onError an onTimeout. They are usually called callback functions.

1.1 Protocol and Factory

Protocol describes the application protocol in the context of Twisted application. Factory is a class definition for server instances or client instances.

To better understand how to build customized protocol and intialize a server, an example is analyzed here.

Example 1

```
from twisted.internet import protocol, reactor, endpoints

class Echo(protocol.Protocol):
    def dataReceived(self, data):
        self.transport.write(data)

class EchoFactory(protocol.Factory):
    def buildProtocol(self, addr):
        return Echo()

endpoints.serverFromString(reactor,
"tcp:1234").listen(EchoFactory())
reactor.run()
```

Example 1 creates a server binding to port 1234 with TCP protocol. The server which is described by EchoFactory. EchoFactory is constructed from class protocol. Factory with protocol Echo. Echo protocol describes what to do when server receive data from client, which is to reply with the same data.

1.2 Code Efficiency

Twisted framework provides users with a good abstraction of its APIs. It greatly speeds up the development of architecture. For instance, you can create storage on server instance by setting a attribute on self.factory.

Example 2

self.factory.port = 12000

self.factory.clients = ['Alice', "Bob", "Carol"]

Although the server storage is ephemeral, it is very convenient for testing new architectures.

Twisted also makes common requests easy, such as http get request. twisted.web.client provides us with getPage method which is equivalent to http get. We can append callbacks to the returned getPage request by using addCallback method.

The powerful API saves us both time and lines to write useful code.

1.3 API Documentation

Twisted documentation is very limited. Many existing APIs are undocumented. See https://twistedmatrix.com/documents/current/api/twisted.internet.protocol.html for an example of poorly maintained documentation.

2. Server Herd Proxy

The server protocol handles three types of requests:

IAMAT handles clients location update, which propagates on a new timstamp to other servers.

WHASAT handles location query for some client with client id. It first make sure such client exists, then it generate a query with that client's location to Google Places API to request for nearby places. The results is limited by radius and number of entries which is part of the WHATSAT request.

AT request is a request sent from server to server for the sake of flooding. Since we want to keep servers updated with the latest client locations, we flood any location update that has a time stamp later than what is on a server for that client. Flooding guarantees each server has the latest location of any client (under the rules of communication provided by the spec). The flooding is guaranteed to stop because the time stamp eventually becomes the same among all servers.

3. Testing

Twisted lacks a built-in testing tool. I implemented a simple logging system with python logging. Testing is facilitated with the help of bash script and Linux aliases

3. Comparison

Twisted provides a bigger variety of protocols than Node.js supports. Based on my research, Node.js is proved to be faster and more scalable than Twisted. The main complaint for Node.js is that promises become hard to track in a big application. The confusing promises chains are nicknamed the “Callback Hell”.

The main difference for me between Twisted and Node.js is popularity. There are many enthusiasts for Node.js whereas Twisted are mostly advocated by experienced and mature programmers. There are more active development in Node.js, which means a faster growth. Node.js also uses npm packages, which is popular among javascript developers.

Both Node.js and Twisted implements event-loop. This also means that they have the best performance when the servers are dealing wih non-blocking I/Os.

4. Running

bash run_herds.sh # runs 5 servers

bash toast_herds.sh #stop 5 servers

bash restart.sh #restart 5 servers

telnet localhost [port]

* Since the public access for Google API is restricted to specific IPs, testing for WHATSAT needs to be done with IP corresponding API key .

5. Reference

"What Are the Use Cases of Node.js vs Twisted?" *Javascript*. N.p., n.d. http://stackoverflow.com/questions/3461549/what-are-the-use-cases-of-node-js-vs-twisted. Web. 02 Mar. 2015.

"What Are the Disadvantages of Using Node.js?" http://nellaikanth.tumblr.com/post/90246998371/what-are-the-disadvantages-of-using-node-js. *The Beautiful Mind* . N.p., n.d. Web. 02 Mar. 2015.

"Developer Guides" *Developer Guides* — Twisted 15.0.0 Documentation http://twistedmatrix.com/documents/current/core/howto/index.html. N.p., n.d. Web. 02 Mar. 2015.

.