Twisted Places Proxy Herd

Alan Covarrubias
103996981

Abstract

Twisted is an asynchronous event-driven networking framework written in Python. I will analyze this framework with respect to its viability as a proxy herd server with automatic place information updating between servers. I will analyze its ease of implementation, maintainability, and reliability and compare these aspects with another event-driven framework Node.js.

1. INTRODUCTION

The goal of this project was to determine how well Twisted could replace the LAMP infrastructure used by Wikipedia in a scenario where updates to articles will happen more often, various protocols will be used to connect with the servers, not just HTTP, and clients will be more mobile. These constraints make LAMP obsolete, which forced us to consider using Twisted as a new networking framework.

2. IMPLEMENTATION

ProxyServerFactory

Our prototype instantiates one server at a time, with a total of up to 5 separate servers in the herd. Each server listens for messages on a certain port, and calls on protocols to handle any events (requests from clients). Every instance of a server is actually an instance of ProxyServerFactory that instantiates a ProxyServerProtocol when an event occurs. A log file is created per server created called server-{server name}.log that records all events that are processed and how they are processed.

ProxyServerProtocol

ProxyServerProtocol inherits from the LineReceiver Twisted class and it handles events sent to the ProxyServerFactory. Every time a message is sent to the ProxyServerFactory, the method lineReceived is invoked. This method parses the command and invokes the following methods depending on the values sent to the server.

IAMAT

When a client sends this request to a server, the contents of the message are parsed, and a new entry is added to the client dictionary stored in the server. This dictionary stores information about the client that sent the IAMAT message, most

importantly the position of the client. It then floods the information of the client to the other servers in the herd using the AT method.

WHATSAT

When a client sends this request to a server, the server looks for the client's location in the client dictionary and sends an asynchronous request to the Google Places API. The Google API takes the position and the radius and returns JSON data specific to that location. When the API request returns, a callback is issued that sends out the JSON data that the API responded with along with the original request.

AT

An AT message is sent back to clients as a respond to an IAMAT request and is also sent from server to server using a ProxyClientFactory that creates ProxyClientProtocols that send messages to other servers when a location in the clients dictionary is updated. This is executed in the flood method. In order to avoid an infinite loop, each server checks any AT message's received time with the last updated time value of the client, and kills any outdated messages.

3. RESULTS

Twisted's features make it a good alternative for LAMP. It's ability to write asynchronous code with the use of callbacks allow for more frequent updates due to the nonblocking features that come with asynchronous code. Twisted also supports a wide variety of protocols including TCP, UDP, HTTP, and others which is a requirement for our prototype. Mobile clients can be handled easily as our prototype implementation shows. Python is also a simple and flexible language which makes coding servers easy.

4. NODE.JS

Node.js uses the same asynchronous programming methodology as Twisted which makes it a viable alternative to Twisted as well. Although Node.js is newer than Twisted, it has become quite popular in a short amount of time. Since it runs on the V8 JavaScript engine, it might even be a better alternative to Twisted since this virtual machine is very robust. Either one of these frameworks would work for the purposes of our application.

REFERENCES

http://krondo.com/an-introduction-to-asynchronous-programming-and-twisted/

https://nodejs.org/en/

https://twistedmatrix.com/trac/