**CS 131 Project: Twisted Places Proxy Herd**

*Wilsen Kosasih – University of California, Los Angeles*

**Abstract**

There are a lot of architectures web-based applications can be based upon. However, in a system where quick updates to the database or view is needed, not all architectures will produce the same result.

To answer the need of the efficiency problem, application server herd was introduced, where multiple application servers communicate directly to each other, propagating update to other servers when there is a change in one of the communicating server.

**Introduction**

In this project, we were tasked to implement an application herd in order to propagate update in a server in a quick manner. This report will describe the implementation of each tasks required by the spec of the project. It will then be followed by the comparison of Python and Java programming language. Also, I will compare the overall approach of Twisted against Node.js. I will end the report with a conclusion whether or not Twisted is a suitable Framework for this project.

1. **Application Implementation**

The application was implemented using a file called “chatserver.py” and a “conf.py” file that specifies the API\_KEY, port numbers of the servers, and neighboring servers. This was done by using a reactor that runs in a infinity and handles clients who try to connect to the server.

Whenever a client tries to connect to the server, the application will thoroughly check whether the arguments is valid or not, such as if the command issued is applicable, if the argument types is correct, or even if the number of arguments passed is correct in accordance to the command issued.

This is done so that data received can be safely propagated to other servers without causing errors.

After checking the validity of the arguments, the application will either produce an error message or transfer the data to the neighboring servers through the use of TCP socket.

Meanwhile, a server log file will be produced to enable the users to inspect the changes made by the server while it was running. Every time the server tries to propagate data or if it catches an error, a message will be produced to the log file.

The server has three commands functionality, which should be separated from each other as much as possible to avoid complications.

These three commands are:

1. IAMAT command
2. WHATSAT command
3. AT command
   1. **IAMAT command**

The IAMAT command is used by a client to tell the server where the client is currently located. The format of the IAMAT command is

IAMAT [client ID] [latitude] [longitude]

It is necessary to check whether the number of arguments is correct or not. It is also important to check if the arguments passed are valid. Specifically, we need to check whether latitude and longitude are of type Float. We also need to check if the user’s timestamp is valid.

If an error is found, the terminal will show a message of “?” followed by a space and the command typed by the client. A line indicating invalid arguments will be outputted to the log file.

If the user passed valid arguments, we then need to create a response message to the client in the form of

AT [Server ID] [Client-Server Time Difference] [client ID] [latitude] [longitude]

The application will then propagate this message to the neighboring servers by the use of reactor.

* 1. **WHATSAT command**

The WHATSAT command is used by a client to ask the server to find nearby places by utilizing Google Place API. The format of the WHATSAT command is

WHATSAT [client ID] [radius] [upper bound on information]

As before, it is necessary to check the number of arguments passed. It is also important to check if the arguments passed are valid. It was distinctively specified that radius must be between 0 and 50 km while the upper bound must be between 0 and 20. We also need to check if the user’s timestamp is valid.

Similar as IAMAT command, If an error is found, the terminal will show a message of “?” followed by the command typed by the client. A line indicating the error will be outputted to the log file.

If the user passed valid arguments, we then handle the arguments by creating a URL to Google Place by using the arguments and API\_KEY. If the HTTP request was successful, we then create a response message to the client in the form of

AT [Client-Server Time Difference] [client ID] [latitude] [longitude]

{

JSON-Format Message

}

* 1. **AT command**

Unlike the other two commands, this command is not being used by the client. Instead, it is produced by the server as a response to the client and also being used to propagate data to the other servers. The format of the command is

AT [Server ID] [Client-Server Time Difference] [client ID] [latitude] [longitude]

Even more so than the other two commands, it is really important to check the validity of the arguments to avoid errors during propagation.

1. **Python vs. Java**

The biggest difference between Java and Python can be said to be Duck Typing. Java is statically typed while Python is dynamically typed. This means that Python is write and read. However, the downside of this is that it might be hard to analyze python code compared to Java one.

Java would often catch the errors that would cause runtime error or a crash in Python during compile-time. Yet, this comes as a price as Java type conversion is pretty limited while Python can be really flexible.

Meanwhile, in memory management, Python seems to have better garbage collector strategy. As soon as there is no reference to an object, it immediately removes the object from memory, which causes Python to optimized memory management.

The downside of using Python is its weak multi-threading. This is because it has to wait for Global Interpreter Lock, causing it to be slower than Java in multithreading. However, in this implementation, we did not need a multithread.

**3. Twisted vs. Node.js**

While there are many similarities between the two, Twisted is an extremely mature API that has been used and experimented on longer than Node.js. As a result, Twisted has much more functionalities than Node.js albeit a bit slower.

Twisted is reader friendly as it uses simple syntax. Meanwhile, Node.js might be harder to work with, especially for beginners.

**4. Conclusion**

Twisted has been proven to be really useful due to its functionalities. However, Node.js, which is a bit faster, might be more suitable in implementing this project, which is highly dependent on speed and efficiency.

**References**

[1] Python & Java: Side-By-Side Comparison. (2009, October 3). Retrieved From

https://pythonconquerstheuniverse.wordpress.com/2009/10/03/python-java-a-side-by-side-comparison/