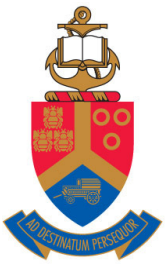

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

NavUP
Longsword Data Implementation

Compiled By

Lucian Sargeant - u15225560
Ritesh Doolabh - u15075754
Peter Boxall - u14056136
Claude Greeff - u13153740
Harris Leshaba - u15312144
Hristian Vitrychenko - u15006442

GitHub Repository: COS 301 Team Longsword Data GitHub Repository(Phase 3)

2017
TEAM LONGSWORD (DATA)

Contents

1 Introduction

The implementation of the Data module involves data streaming as its prime objective. This is to ensure that data is transmitted between devices at a rate that will not bottleneck the application as well as being able to handle a large amount of requests concurrently.

1.1 Technology Choices

To ensure that our objectives were met, two important technology choices are Flink and Aruba. Flink is a data streaming service by Apache that provides data streaming to process multiple requests concurrently. Flink is optimized for cyclic or iterative processes by using iterative transformations on collections. This is achieved by an optimization of join algorithms, operator chaining and reusing of partitioning and sorting. However, Flink is also a strong tool for batch processing which was used to test the speed of the system. Flink streaming allows the processing of data streams as true streams, i.e., data elements are immediately "pipelined" though a streaming program as soon as they arrive. This allows to perform flexible window operations on streams. It is even capable of handling late data in streams by the use of watermarks. Furthermore Flink provides a very strong compatibility mode which makes it possible to use other software in conjunction with it, for example Kafka.

The second technology choice is Aruba. This is a system implemented by the University Of Pretoria that uses the WiFi access point to allow digital devices to be located by the use of their MAC Addresses. Flink primarily communicates with the interface sending the MAC Address and queries the Aruba server to retrieve the location of the device and returns it to the calling program.

2 Running The Program

The entirety of the submitted demo program has three running modes. These are namely:

- Flink only - to demonstrate streaming capabilities.
- Aruba only - to demonstrate Aruba communication, requesting and receiving of location data.
- Live requesting - to demonstrate the real time requests and responses from using Aruba and Flink.

2.1 Flink Only

1. Navigate to flink-1.2.0/bin
2. Open a command terminal and run start-local.bat/.sh
3. Run TestServerRequester with arguments [false]
4. Run TestServerReceiver with arguments [false false]
5. Upload Data.jar to flink by running ./bin/flink run ./examples/Data.jar

2.2 Aruba Only

1. Navigate to flink-1.2.0/bin
2. Open a command terminal and run start-local.bat/.sh
3. Run TestServerRequester with arguments [true]
4. Run TestServerReceiver with arguments [true false]
5. Upload Data.jar to flink by running ./bin/flink run ./examples/Data.jar

2.3 Live Requesting

1. Start a netcat terminal on port 9000 with the command: nc -l -p 9000
2. Start TestServerReceiver with arguments [true true]
3. Upload Data.jar to flink by running ./bin/flink run ./examples/Data.jar
4. Enter any MAC Address into netcat (output will be displayed in TestServerReceiver)