Ribbon

Objectives:

- Literature review of Ribbon

- Implementation of a working prototype for Ribbon

- Integration of Ribbon into the Internet of Things (IoT) project.

References

Appendixes

Literature Review of Ribbon

Introduction:

Ribbon is part of the Netflix OSS (Open Source Software) library offering, providing a robust, resilient and intelligent inter-process and service communication for the cloud platform.

The basic functionality of Ribbon is to work as a software *load balancer* for communication between group of servers and their clients. This is achieved by either providing DNS and/or IPaddress of communicating servers or rotating among a list of server according to certain logic.

The default logic is *RoundRobbinRule*, where servers are selected in a sequence depending on failure of current connection.

Other rules are *AvailabilityFilteringRule* and *WeightedResponseTimeRule*.

Known servers can be listed in either using an API or as a default in the configuration file following the format:

*<client-name>.ribbon.listOfServers=www.abc.com:80,www.def.com:80*

Advance configuration are possible for load balancing by using server list filter properties by either returning zone related servers or a subset of servers depending on their performance (e.g. availability, number of connection currently held, etc.).

Implementation of a Working Prototype for Ribbon

Scenario: A Ribbon client interacting with three microservices.

Ribbon Client:

The Ribbon Client acts as a consumer for the microservice. The code is given in the Appendix A.

Microservices:

The Microservices act as a server for the client. Hence the same name will apply to them, though the port numbers are different (when running locally).

Ribbon Client

app-name: client

port: 8991

MS1

app-name: MS

port: 8992

response: ford

MS2

app-name: MS

port: 8993

response: fruit

MS3

app-name: MS

port: 8994

response: cat

Default Round Robin Rule