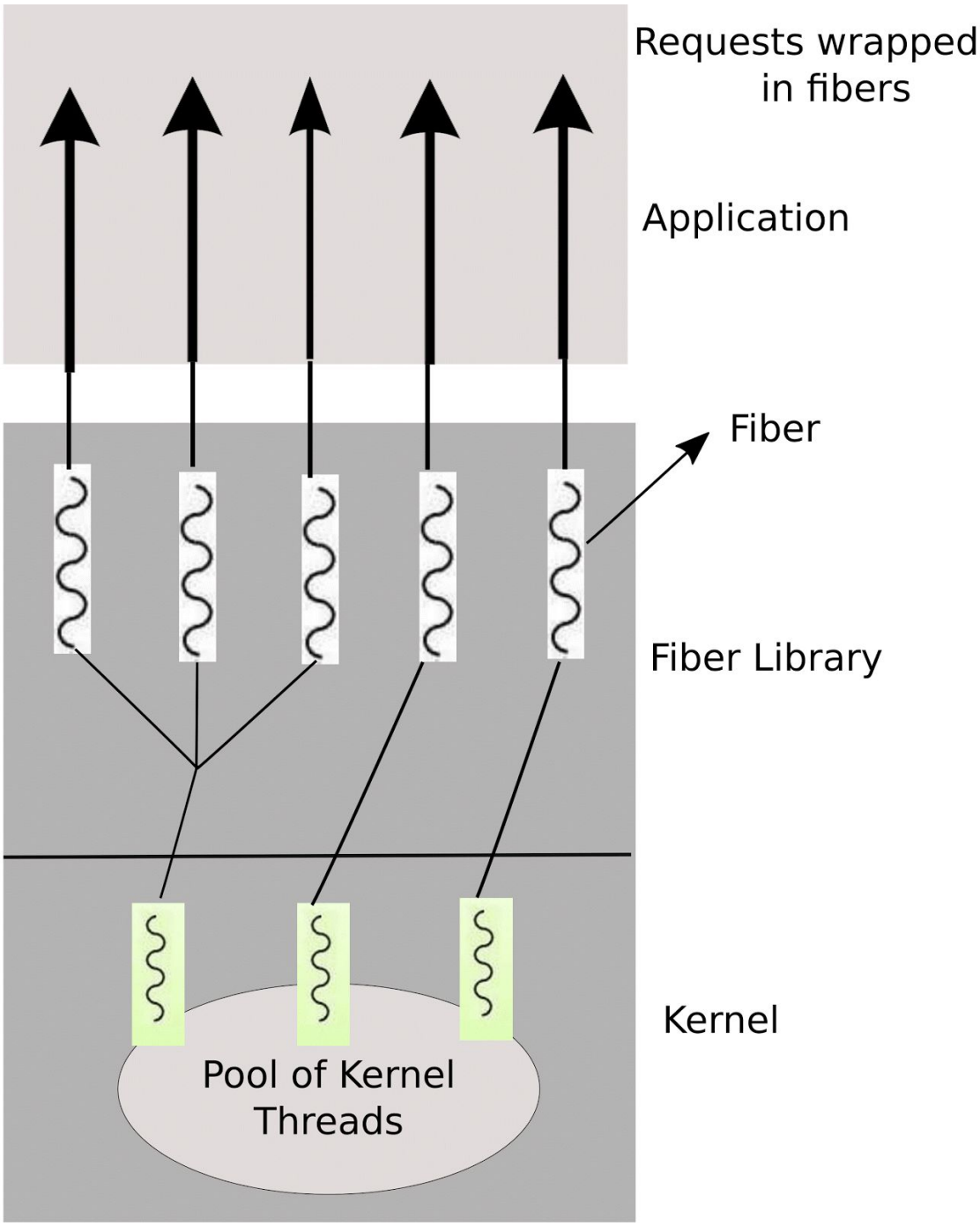
LoadGen 1.0

Introduction

LoadGen is a web application which is designed to load test client/server software (such as a web application). It is developed based on fibers which is a lightweight thread implementation in Java. This implementation helps in generating very high load without consuming much resources. It can be used for HTTP, database and web service testing.



Quasar core is a Java library which implements lightweight threads which can run on JVM. It is called as fibers. Fibers can be instantiated and run just like regular threads. The above figure shows how LoadGen generate requests. In LoadGen, there will be a fixed number of kernel threads to which the user level threads or the fibers gets scheduled. The fiber library will handle the scheduling. The requests will be wrapped inside the fiber and sent.

Obtaining the code

- Download the code from github. https://github.com/stanlyjohn2/LoadGenerator
- Build the code using eclipse by importing the downloaded project.
- Read documentation for user guide and developer guide.

LoadGen directory structure

- src - contains the source code with java, jsp, css and js files.
- docs- contains the user documentation and developer's guide.
- Pom.xml- contains the instructions to install dependencies and libraries needed for the project

src directory structure :

- src/main/webapp/resources/js contains the javascript files to handle the UI interface and to call the backend functions.
- src/main/webapp/resources/images contains all the image resources used in the web application.
- src/main/webapp/resources/css contains the css files for UI interface.
- src/main/webapp/WEB-INF/views/jsp have all the webpages used in the application.
- src/main/java/com/webQ/controller contains the functionalities which will handle the calls from the front end of the web application.
- Running fiber inside functions.

  Ex:

```
Fiber<Void> testplansfiber = new Fiber<Void>(() -> {
        currtestplan.execute(null, normaltest);
}).start();
```