## 面向交互式Web应用的负载生成工具设计与实现

## Introduction

测试工具的重要性。

JavaScript是一个动态类型、基于对象的脚本语言 ，需要JavaScript执行引擎来执行程序 。目前JavaScript被广泛使用在Web应用中，允许在客户端动态加载网页内容。Web2.0应用的一个重要特性是使用JavaScript和Ajax技术，增强了与用户之间的交互性。JavaScript已经成为Web应用的一个重要部分，在Alexa最受欢迎网站排名前100的网站中，几乎所有网站都包括JavaScript方法。

由于一个Web应用常常是大量用户同时使用，人工测试无法完成大量用户的并发请求，因此必须借助自动化测试工具。这些工具可以模拟用户行为，向服务器发送和接收请求，并且获得服务器的响应时间及吞吐量等信息。但是这些工具具有一些局限性，它们不测试JavaScript和Ajax的执行行为，因此，无法模拟真实的Web2.0应用负载的特性，也无法得到它们对Web服务器及响应时间等性能的影响。

随着云计算的发展，越来越多的Web应用部署在云平台，根据用户需求动态提供资源，因此，对Web应用的弹性测试也是非常重要的一个方面，需要利用负载生成工具生存动态负载。

在这篇论文中，我们选择Faban这个开源的Workload生成及执行框架进行扩展，主要的工作有以下三点：

1. 使用JavaScript脚本的执行引擎，实现执行Web应用中的JavaScript和Ajax操作，并搜集这些操作的时间、结果等信息。
2. 分析Web页面中的元素，根据给定的模式进行模拟用户的行为，实现操作的自动化执行。
3. 用户给定一个函数，针对某个Web应用，可生成符合该函数形状的Workload。改进在分布式部署情况下，Faban的动态负载生成功能，使得动态负载文件自动分配到各个agent中。

## Related Work

目前已有一些针对Web2.0应用的负载生成工具，Faban是一个开源的Workload生成及执行框架，提供一个马尔可夫链式的、闭环的、基于会话的Workload生成器，可以模拟上千用户的操作行为。但是Faban不提供测试JavaScript和Ajax的执行行为的功能，而且用户需要针对每个请求操作手动编写测试代码。 Faban在运行的过程中可以根据指定的文件，来改变模拟的用户数量，可是这个文件需要用户手动放在每台agent机器上，增加了用户的工作量。

JavaScript是一个动态类型、基于对象的脚本语言，需要JavaScript执行引擎来执行程序。V8 是 Google 发布的开源 JavaScript 引擎，采用 C++ 编写，在 Google 的 Chrome 浏览器中被使用。Rhino是用Java编写的JavaScript引擎的开放源代码实现，是Mozilla项目的一部分。它最常被用于嵌入Java应用程序，以便为终端用户提供脚本的能力。

HtmlUnit是JUnit的开源扩展测试框架之一，可以模拟浏览器的行为，将HTML文档抽象为对象。它使用Rhino引擎，能够解析并执行页面上的JavaScript代码，模拟JavaScript运行，并且支持主流的JQuery类库。HtmlUnit使用Java开发，提供丰富的API，有较好的扩展性。通过对HtmlUnit框架进行扩展，模拟用户表单填写、点击等行为， 实现页面中JavaScript代码的执行，搜集Web请求的响应时间、返回结果等数据。

## Motivating Example

## Overview

## Experiment

## Future Work