funkload通用脚本使用方法

作者	审核	版本	日期	备注
严嘉蔚	刘勇	1.0	2014-07-14	初稿

1. funkload安装

1、bubuntu下执行命令

sudo aptitude install python-dev python-setuptools python-webunit python-docutils gnuplot

sudo aptitude install tcpwatch-httpproxy --without-recommends

sudo easy_install -U funkload

装完后是最新的稳定版本，如果想安装最新的线上版本，将最后一行替换成sudo easy_install -f http://funkload.nuxeo.org/snapshots/ -U funkload

2. 通用脚本下载地址

http://192.168.10.10/svn/QA/auto_test/apollo/automation/performance/

3. 使用方法

1，解压缩脚本

2，修改FunkloadCommon.conf文件

3，通过funkload命令行执行测试脚本

4，生成report页面

3.1. 配置文件说明

FunkloadCommon.conf为funkload脚本的配置文件，"[ ]"内的是一个域名，每个域下面会有若干个配置。

3.1.1. main域

url

url为需要测试的服务器的ip地址和端口号，不包含路径，最后的“/”也不要包含

例：url=http://192.168.1.48，url=http://192.168.1.48:8080，http://www.baidu.com

ok_log

页面返回为期望值时将响应的body输出到该文件

例：ok_log=/tmp/funkload_log

ok_log_write

正常的log是否需要打印，该配置不为true则不输出到上述文件，只打印在终端里

例：ok_log_write=true

error_log

当脚本执行发生一场，包括系统异常或者页面返回非期望值后异常log输出到该文件

例：error_log=/tmp/funkload_errorlog

3.1.2. pages域

page_number

需要测试的页面数，每个页面的属性在[page_x_section]域中配置，x表示页面编号，从0开始。[page_x_section]的配置详见下文。

如果page_number的值大于[page_x_section]域的个数脚本会报错，如果page_number的个数小于[page_x_section]则只测试前page_number个页面

test_page_list

需要测试的页面。该配置有三种写法，all，noset或者[0, 1, 3]

all表示所有页面均测试，例如page_number=4的话脚本就会分别读取[page_0_section]，[page_1_section]，[page_2_section]，[page_3_section]的配置并进行测试，此时[page_x_section]中的need_test字段不起作用

noset表示不指定测试页，由[page_x_section]中的need_test字段决定该页面是否要测试

[0, 1, 3]表示需要执行第一个，第二个和第四个页面，但如果page_number配置为3，则对第一个和第二个页面进行测试，第四个无法遍历到所以不测

3.1.3. page_x_section

x代表page编号，从0开始，加入page_number为4，则需要4组page_x_section，分别为page_0_section，page_1_section， page_2_section， page_3_section，否则就会出错，以此类推。

need_test

该配置表示当前页面是否需要测试，如果为true为测试，false为不测试。该参数只有在test_page_list选项为noset时才起做用，否则以test_page_list中指定的为准

page_path

当前页面的url路径，不包含ip和端口号，可以带参数?xxx=xxx

例：要测试百度的搜索页面 http://www.baidu.com/s?wd=a

则在main域的url=http://www.baidu.com，而此处的page_path=/s?wd=a

page_method

http请求方法，可以是GET，POST，PUT，DELETE

headers

http请求时头参数，该配置为可选

配置方法为key1===value1&&&key2===value2，以此类推

例：某项目的借口发送请求时需要带头信息，有3个参数，分别为

X-Thunder-Client：some-pirate-client

X-Download-Protocol：http

X-Download-Type：mp4

则headers=X-Thunder-Client===some-pirate-client&&&X-Download-Protocol===http&&&X-Download-Type===mp4

request_type

http请求的类型：可选参数None，data，params三种

当http请求的request_body为空时使用None

如果request_type为params时，下面阐述的data_type为None，data为[["key1", "value1"].[“key2”,“value2”]]这样的类型

如果request_type为data时，表示一个http请求包含了request body，而request body通常是一串数据，可以是字符串，可以一段json数据，甚至是一串数据，比如pyweb的ingest接口上传文件并非使用http上传数据所使用的multipart/form-data数据结构，而是直接在post接口请求中将文件的二进制数据以request_body发送过去。

data_type

如果request_type为None或者是params时data_type为None

如果request_type为data时，data_type根据请求情况做配置，可以是text/html，text/xml，text/json等等，这个类型在发送http请求时作为类型，理论上是一个标记

data

http请求数据，如果request_type为params时以参数形式发送格式为[["key1", "value1"].[“key2”,“value2”]]。事实上当page_method为GET时请求一般不会包含request_body。所以例如如果某个page的配置为url=http://www.baidu.com，page_method=GET，page_path=/s，request_type=params，data=[["wd", "a"], ["rsv_spt", "1"], ["issp", "1"]]和page_path=/s?wd=a&rsv_spt=1&issp=1，request_type=None，data=None效果是一样的。但是如果page_path=/s?wd=a，request_type=params，data=[["rsv_spt", "1"], ["issp", "1"]]，实际请求的url是http://www.baidu.com/s?wd=a?rsv_spt=1&issp=1，就会出错。事实上GET请求data为[["key1", "value1"].[“key2”,“value2”]]脚本就是吧path + "?" + key1=value1&key2=value2作为完整的url进行请求

post方法一般包含request body，如果想发送一段字符串如asdfghj则data=asdfghj，如果需要发送一段json，而json是保存在一个文件里，比如/tmp/json，则data=@/tmp/json，@开头的配置脚本会认为是个文件而去读取该文件。上述两种情况request_type均为data。如果想以http标准的上传文件协议，即multipart/form-data格式，则request_type使用params，data可以写成[["video", "upload@/tmp/xxx.dna"], ["duration", "20"]]，以upload@开头的配置系统会处理成标准multipart/form-data格式上传文件

ok_codes

测试通过的return code

一般情况下http，200，30X表示是正常的返回，40X，50X为错误相应，如果，ok_codes为None，则当请求为200，30X为通过，40X，50X报错，但是某些测试希望40X，50X也作为正常响应则ok_codes=200,400,404。除此之外的范围均抛错，分割的逗号是英文的逗号。切记当如果ok_codes为自定义的时候，如果200忘记添加了，比如ok_codes=400,404则返回200也是会抛错的。

4. funkload测试执行

4.1. 单一运行脚本

cd到脚本目录执行命令fl-run-test -dv common.py执行完后在脚本目录下的simple目录下生成一个simple-test.log文件，如果想要修改名字或者路径，可在FunkloadCommon.conf的ftest域中修改

4.2. 并发测试

cd到脚本目录执行命令fl-run-bench -c 1:10:4:8 -D 60 common.py FunkloadCommon.test_common则脚本会分别执

行4轮，第一轮1个并发，FunkloadCommon.conf的bench域修改

命令行测试时可不指定-c和-D，如果不指定-c和-D则脚本会读取bench域的cycles和duration来配置并发数和时间

4.3. 生成报告

cd到脚本目录执行fl-build-report --html bench/bench-test.xml，会在脚本目录下生成一个报表文件夹，进入后打开里面后使用浏览器打开里面的index.html文件就能看到详细的报告