mvn gatling:test -Dgatling.simulationClass=blazedemo.BlazeDemoDisableRedirect

mvn gatling:test -Dgatling.simulationClass=blazedemo.BlazeDemoDonotShowResourcesInReport

mvn gatling:test -Dgatling.simulationClass=blazedemo.BlazeDemoShowResourcesInReport

mvn gatling:test -Dgatling.simulationClass=blazedemo.BlazeDemoGroupShowResourcesInReport

mvn gatling:test -Dgatling.simulationClass=blazedemo.BlazeDemoOneLoginLogoutMultipleOperation

mvn gatling:test -Dgatling.simulationClass=blazedemo.

JsonResponseSaveKEYValue

mvn gatling:test -Dgatling.simulationClass=blazedemo.TryMaxException

mvn gatling:test -Dgatling.simulationClass=blazedemo4.BlazeDemo6

mvn gatling:test -Dgatling.simulationClass=blazedemo4.RegSimAtOnceUsers

mvn gatling:test -Dgatling.simulationClass=blazedemo4.RegSimConstantUsersPerSec

mvn gatling:test -Dgatling.simulationClass=blazedemo4.RegSimConstantUsersPerSecRandomized

mvn gatling:test -Dgatling.simulationClass=blazedemo4.RegSimrampUsersDuring

mvn gatling:test -Dgatling.simulationClass=blazedemo4.RegSimRampUsersPerSecRate1Rate2

mvn gatling:test -Dgatling.simulationClass=blazedemo4.ClosedModel

mvn gatling:test -Dgatling.simulationClass=blazedemo4.ClosedModelIncrementConcurrenUserseachLevelLasting

mvn gatling:test -Dgatling.simulationClass=blazedemo4.ClosedModelrampConcurrentUsersFromToDuring

mvn gatling:test -Dgatling.simulationClass=blazedemo4.ClosedModelRampUprampConcurrentUsersFromToDuringAndRampDown

Complete Code

package blazedemo4  
import io.gatling.core.Predef.\_  
import io.gatling.http.Predef.\_  
  
class RegSimAtOnceUsers extends Simulation {  
 val *domain* ="blazedemo.com"  
 var *csvFedderBlazeDemoFilightDetails* = csv("data/BlazeDemoFilightDetails.csv").circular  
  
 val *httpProtocol* = http  
 .baseUrl("https://" + *domain*)  
 .inferHtmlResources(BlackList(""".\*\.js""", """.\*\.css""", """.\*\.gif""", """.\*\.jpeg""", """.\*\.jpg""", """.\*\.ico""", """.\*\.woff""", """.\*\.woff2""", """.\*\.(t|o)tf""", """.\*\.png""", """.\*detectportal\.firefox\.com.\*"""), WhiteList())  
 .acceptHeader("text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,\*/\*;q=0.8,application/signed-exchange;v=b3;q=0.9")  
 .acceptEncodingHeader("gzip, deflate")  
 .acceptLanguageHeader("en-GB,en;q=0.9,en-US;q=0.8,mr;q=0.7,hi;q=0.6")  
 .contentTypeHeader("application/x-www-form-urlencoded")  
 .originHeader("https://blazedemo.com")  
 .upgradeInsecureRequestsHeader("1")  
 .userAgentHeader("Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_15\_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/98.0.4758.102 Safari/537.36")  
  
 val *headers\_0* = *Map*(  
 "pragma" -> "no-cache",  
 "sec-ch-ua" -> """ Not A;Brand";v="99", "Chromium";v="98", "Google Chrome";v="98""",  
 "sec-ch-ua-mobile" -> "?0",  
 "sec-ch-ua-platform" -> "macOS",  
 "sec-fetch-dest" -> "document",  
 "sec-fetch-mode" -> "navigate",  
 "sec-fetch-site" -> "same-origin",  
 "sec-fetch-user" -> "?1")  
  
 *// FromPort,ToPort,Airline,Price,FlightNo* object Reserve{  
 def methodReserve ={  
 exec(http("T1 Reserve Page")  
 .post("/reserve.php")  
 .headers(*headers\_0*)  
 .formParam("fromPort", "${FromPort}")  
 .formParam("toPort", "${ToPort}"))  
 *//.pause(5)* }  
 }  
   
 object Purchase{  
 def methodPurchase ={  
 exec(  
 http("T2 Purchase Page")  
 .post("/purchase.php")  
 .headers(*headers\_0*)  
 .formParam("flight", "${FlightNo}")  
 .formParam("price", "${Price}")  
 .formParam("airline", "${Airline}")  
 .formParam("fromPort", "${FromPort}")  
 .formParam("toPort", "${ToPort}"))  
 *//.pause(2,5)* }  
 }  
  
 object Confirmation{  
 def methodConfirmation ={  
 exec(http("T3 Confirmation Page")  
 .post("/confirmation.php")  
 .headers(*headers\_0*)  
 .formParam("\_token", "")  
 .formParam("inputName", "Testing")  
 .formParam("address", "Address1 ")  
 .formParam("city", "NY")  
 .formParam("state", "Ney york")  
 .formParam("zipCode", "12345")  
 .formParam("cardType", "visa")  
 .formParam("creditCardNumber", "1234567890")  
 .formParam("creditCardMonth", "12")  
 .formParam("creditCardYear", "2017")  
 .formParam("nameOnCard", "Automation Testing")  
 *//.check(bodyBytes.is(RawFileBody("blazedemo4/blazedemo4/0002\_response.html")))* )  
  
 }  
 }  
  
val *scnRefactor*= scenario("Fligh Booking" )  
 .during(60){  
 feed((*csvFedderBlazeDemoFilightDetails*))  
 .exec(Reserve.*methodReserve*)  
 .exec(Purchase.*methodPurchase*)  
 .exec(Confirmation.*methodConfirmation*)  
 }  
  
 setUp(*scnRefactor*.inject(  
 atOnceUsers(10))  
  
 ).protocols(*httpProtocol*)  
  
}

Content of file “/gatling-maven-plugin-demo-main/src/test/resources/data/BlazeDemoFilightDetails.csv”

*FromPort*,*ToPort*,*Airline*,*Price*,*FlightNo  
Paris*,*Rome*,*Virgin America*,*472.56*,*43  
Paris*,*Rome*,*United Airlines*,*432.98*,*234  
Paris*,*Rome*,*Aer Lingus*,*200.98*,*9696  
Paris*,*Rome*,*Virgin America*,*765.32*,*12  
Paris*,*Rome*,*Lufthansa*,*233.98*,*4346  
Paris*,*London*,*Virgin America*,*472.56*,*43  
Paris*,*London*,*United Airlines*,*438.98*,*234  
Boston*,*New York*,*Virgin America*,*472.56*,*43  
Boston*,*New York*,*United Airlines*,*432.98*,*234  
Boston*,*Berlin*,*Lufthansa*,*233.98*,*4346*

Open Model

/gatling-maven-plugin-demo-main/src/test/scala/blazedemo4/RegSimAtOnceUsers.scala

1

setUp(*scnRefactor*.inject(  
 atOnceUsers(10))  
  
).protocols(*httpProtocol*)

\

Table

Description automatically generated

setUp(  
 *scnRefactor*.inject(  
 *//atOnceUsers(10))* rampUsers(20) during (20)  
)  
)  
 .protocols(*httpProtocol*)

2

Number of Users 20, during 20 second

= every second 20/20 =1 user will logged in for 20 seconds

Total 20 users will logged into system at different time during 1 -20 second period

Started before 19:14:34

At 19:14:42 , half duration time, half of users 20/2 are in system

At 19:14:52, all 20\*20 =400 users are present

At 19:15:53. O users in system 60 seconds after 19:14:52

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

/Users/kzanjad/Documents/Training/SapientTraining/gatling-maven-plugin-demo-main/target/gatling/regsimrampusersduring-20220226134432007/index.html

constantUsersPerSec(20) during (20)

3

Every second 20 users will logged

So total 20\* 20 =400 users will logged into syste during 0-20 second

At 19:11:26 it started

At 19:11:45 all 400 users are working

At 19:12:45 0 users in system 0 . exactly after 1 minute

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

file:///Users/kzanjad/Documents/Training/SapientTraining/gatling-maven-plugin-demo-main/target/gatling/regsimconstantuserspersec-20220226134123922/index.html

* constantUsersPerSec(rate).during(duration).**randomized**:
* Injects users at a constant rate, defined in users per second, during a given duration. Users will be injected at randomized intervals.

4

constantUsersPerSec(20).during (20).randomized

We exepcet total 20\*20 = 400 users after 20 seconds

Every second, random users like some times 25, 26, 18 users are added during first second, second second, tthird second,

After three seconds 69 users are in system

By end of 20 seconds 400 users are in system.

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

file:///Users/kzanjad/Documents/Training/SapientTraining/gatling-maven-plugin-demo-main/target/gatling/regsimconstantuserspersecrandomized-20220226140157833/index.html

rampUsersPerSec(rate1).to.(rate2).during(duration): Injects users from starting rate to target rate, defined in users per second, during a given duration. Users will be injected at regular intervals.

5

rampUsersPerSec(5).to(20).during(20)

/Users/kzanjad/Documents/Training/SapientTraining/gatling-maven-plugin-demo-main/target/gatling/regsimrampusersstartratetargetrateduring-20220226141354696/index.html

Total Users in syste = Avg(5+20) \*20 =

25/2 \*20 =250 users

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, application

Description automatically generated

Closed Model

1

constantConcurrentUsers(20).during(20),

constantConcurrentUsers(nbUsers).during(duration): Inject so that number of concurrent users in the system is constant

Graphical user interface, text, application

Description automatically generated

<file:///Users/kzanjad/Documents/Training/SapientTraining/gatling-maven-plugin-demo-main/target/gatling/closedmodel-20220226144447764/index.html>

2

rampConcurrentUsers(fromNbUsers).to(toNbUsers).during(duration): Inject so that number of concurrent users in the system ramps linearly from a number to another

rampConcurrentUsers(fromNbUsers).to(toNbUsers).during(duration)

rampConcurrentUsers(0).to(20).during(20)

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

file:///Users/kzanjad/Documents/Training/SapientTraining/gatling-maven-plugin-demo-main/target/gatling/closedmodelrampconcurrentusersfromtoduring-20220226145033563/index.html

3

setUp(  
 *scnRefactor*.inject  
 (  
  
 rampConcurrentUsers(0).to(20).during(20),  
 constantConcurrentUsers(20).during(20),  
 rampConcurrentUsers(40).to(0).during(20),  
)  
)  
 .protocols(*httpProtocol*)

Total User should be in system 20 +20

provided during first 60 during loop some users have not log out

total Duration for Simulation = 2 mintures

During loop =60

20

20

20

Total 2 minutes

Userslogged during first 20 second, will starte log out after one minute, during 20:28:20 – 20:28:40, these 20 uses will log out.

Graphical user interface, chart, line chart

Description automatically generated

Graphical user interface, application

Description automatically generated

file:///Users/kzanjad/Documents/Training/SapientTraining/gatling-maven-plugin-demo-main/target/gatling/closedmodelrampuprampconcurrentusersfromtoduringandrampdown-20220226145717033/index.html

rampConcurrentUsers(0).to(20).during(20),  
constantConcurrentUsers(20).during(20),  
rampConcurrentUsers(20).to(0).during(20),

Graphical user interface, application

Description automatically generated

<file:///Users/kzanjad/Documents/Training/SapientTraining/gatling-maven-plugin-demo-main/target/gatling/closedmodelrampuprampconcurrentusersfromtoduringandrampdown-20220226150927011/index.html>

* generate a closed workload injection profile
* with levels of 10, 15, 20, 25 and 30 concurrent users
* each level lasting 10 seconds
* separated by linear ramps lasting 10 seconds
* incrementUsersPerSec is for open workload and
* incrementConcurrentUsers is for closed workload (users/sec vs concurrent users).

setUp(  
 *scnRefactor*.inject  
 (  
 incrementConcurrentUsers(5)  
 .times(5)  
 .eachLevelLasting(10)  
 .separatedByRampsLasting(10)  
 .startingFrom(10) *// Int* )  
)  
 .protocols(*httpProtocol*)

Graphical user interface, line chart

Description automatically generated

Graphical user interface, chart, line chart

Description automatically generated

Graphical user interface, chart, line chart

Description automatically generated

Graphical user interface, chart, line chart

Description automatically generated

Graphical user interface, chart, line chart

Description automatically generated

Graphical user interface, chart, line chart

Description automatically generated

If duratioh is bigger than (10+10)5= 100

Keep duration as 120 seconds

Graphical user interface, chart, line chart

Description automatically generated

file:///Users/kzanjad/Documents/Training/SapientTraining/gatling-maven-plugin-demo-main/target/gatling/closedmodelincrementconcurrenuserseachlevellasting-20220226151608525/index.html

1. How Gatling can send multiple requests or parallel requests during runtime?
2. How to capture both key and value of a key-value pair using correlation?
3. How to do correlation for redirection urls?
4. While recording a HTTPS site, user got certificate error on browser. How to fix this issue to enable recording.

A Try exception

B Report without ramp up and ramp down period

* Capture results only for steady state from simulation log file.
* exclude ramp up and ramp down filter and just show steady state

C

Login once per thread

                   All other activitis.  -5 minutes.  8 hours

Logout

D

Integration with Grafana / Kibana