12/09/2023-Introduction to Performance Testing & Jmeter

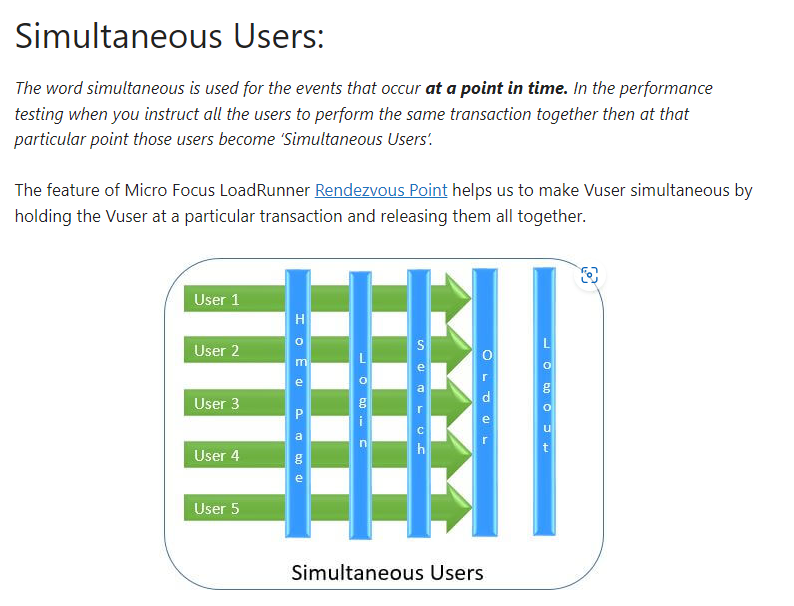
what is performance testing?

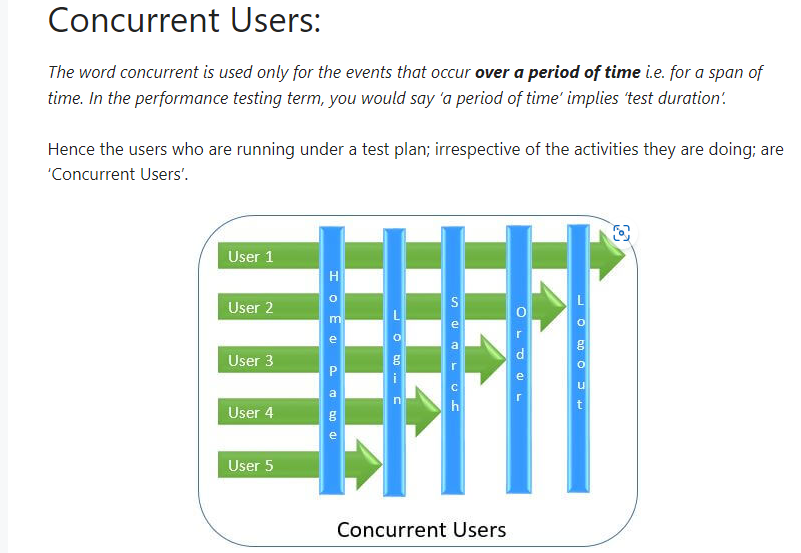
To predict no of user access the application'

To check stability, scalability and response time of the application

Types of test ;

* load test- Application behavior under user load run for 1HOUR.
* Soak Test - Application behavior under user load run for 8hours- It is done to check memory leak.
* Spike Test- Application behavior under sudden user load-beyond break point
* Stress Test- To check the maximum user can access the application-
* Volume Test - To do test for long period





13/09/2023 - Requirement Checklist, Information Gathering & Status Tracking

Functional Testing - It is end to end testing -check all functionality of the application

Eg: Manual testing

non- fun->It is to check the non-functionality like-performance, usability, reliability, Response time

**Requirement checklist**-in excel document.

We have drafted format for requirement checklist which contains all the pre-requisite information required for performance testing

Questionaries document sent to app team (Developer Team )- Request load test on app

**MODE OF COMMUNICATION:**

1. Developer (app) Team send us the mail where they request for doing performance testing for the respected application
2. We replayed them with the pre-requisite check list Excel shell with them and ask them to fill all mandatory details required for performance test
3. If any details missing we ask them to fill the same
4. If app in production - ask through put

**Prerequisite - req checklist Contains:**

* access to app
* bit backet
* repository
* cost tag
* post man collections
* input data/input

**Data requirements**

**AP**I-

* URL,
* Input data (Test data/input data/input- All are same),
* Method.
* Throughput,
* Response time,
* Header,
* Body/payload

**web-app** -

* URL,
* scenario,
* Input data (Test data/input data/input- All are same),
* Throughput,
* Response Time

**TO CALCULATE THROUGHPUT**:

Through Put=hits/sec = No.Of.Request

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Response time in seconds(1 hour = 3600 seconds)

14-09-2023 - API - Postman Tool, JMeter scripting

Difference between Web applications and API

Web\_app - There is direct interaction between UI and user

API -3rd party needed to access it.

**3-Tier architecture**

* Presentation tier- UI layer- where we able to see the output
* Application tier- Where the coding and back-end process where happen. It is like heart for the app
* Database tier - To Store data

**Post man** -

Tool from where we import data for API testing.

API's where first tested manually in Post Man if it works fine then we go for JMeter Scripting.

15-09-2023 - API - Test execution, Response message and Code

* get- Fetch the data
* post-To add new data
* put- Replace the existing data
* patch - Add something new additional to the existing data
* delete- To remove value from the server

18-09-2023 - KT cancelled

19-09-2023 - Response message and Code

200 OK The request is OK (this is the standard response for successful HTTP requests)

* 201 Created The request has been fulfilled, and a new resource is created
* 202 Accepted The request has been accepted for processing, but the processing has not been completed
* 203 Non-Authoritative Information The request has been successfully processed, but is returning information that may be from another source
* 204 No Content The request has been successfully processed, but is not returning any content
* 205 Reset Content The request has been successfully processed, but is not returning any content, and requires that the requester reset the document view
* 206 Partial Content The server is delivering only part of the resource due to a range header sent by the client

300- redirection-kind of sucess -only n web application

* 300 Multiple Choices A link list. The user can select a link and go to that location. Maximum five addresses
* 301 Moved Permanently The requested page has moved to a new URL
* 302 Found The requested page has moved temporarily to a new URL
* 303 See Other The requested page can be found under a different URL
* 304 Not Modified Indicates the requested page has not been modified since last
* requested 306 Switch Proxy No longer used
* 307 Temporary Redirect The requested page has moved temporarily to a new URL
* 308 Resume Incomplete Used in the resumable requests proposal to resume aborted PUT or POST requests

400-user/human/client- Error

* 400- bad request/unauthorized - by user - Error in script- the req cannot be fulfilled due to bad syntax
* 403- forbidden token- incorrect token-request is legal but server refusing to respond to it
* 401- unauthorized request- authentication is not provided. request is legal but server refusing to respond to it
* 405- if we use incorrect API method
* 404- file not found-Wrong URL-They may change environment

500- server error

* 501-internal service error - server doesn’t recognize the req method
* 502- bad gateway -The server was acting as a gateway or proxy and received an invalid response from the upstream server
* 503- dead lock- Service Unavailable The server is currently unavailable (overloaded or down)
* 504- gateway timeout- The server was acting as a gateway or proxy and did not receive a timely response from the upstream server
* 505 HTTP Version Not Supported The server does not support the HTTP protocol version used in the request
* 511 Network Authentication Required The client needs to authenticate to gain network access

|  |  |  |
| --- | --- | --- |
| 200 | Success |  |
| 300 | Redirection |  |
| 400 | Bad Request | Wrong user ID or Password |
| 401 | Unauthorized | Server Issue |
| 402 | Payment Required |  |
| 403 | Forbidden | Wrong Token |
| 404 | Not Found | Wrong URL |
| 405 | Method Not Allowed | Wrong API - Method |
| 500 | Internal Server Error | Server errror |
| 501 | Not Implemented |  |
| 502 | Bad Gateway |  |
| 503 | Service Unavailable | Dead Lock |
| 504 | Gateway Timeout | Timeout |
| 505 | HTTP Version Not Supported | HTTPS Wrong version |

20/09/2023 - WebApp - Blazemeter Plugin, Recording

before recording always clear your browsing history and cache and cookie

click blaze meter extension

naming- Orange HRM-web app-date(20/09/2023)

21/09/2023-WebApp - Secure Auth, Parameterization, Input data

correlation- capture the dynamic value from server when request hit server, the session ID will generate and they where captured in the variable name and passing it in subsequent request using extractors

* pre processor - giving/passing the value before they execute the request
* post processor - giving/passing the value after they execute the request

**Types of extractor:**

x-path extractor ->

//input[@type="hidden"][@name="\_\_RequestVerificationToken"]/@value

syntax: //tag name [@ attribute name]/@attribute name which you need to capture

Json extractor ->

Name of variable: user 1 ; user 2

Json exp; $..variableName (to capture two values in one syntex $..['username','id'])

Match.No: 1 (-1 Takes all the value stored in that name->user1-gokul,raj,mohan )

Default value: Match Not Found

CSS selector expression ->

input[@id=\_\_VIEWSTATE]

syntax: tagname[@attribute name]

Attribute -> value (from which variable name value is passing)

Boundary extractor -> abysiuhbyugfyj"abcd"gytfytguihi

Left Boundary : abysiuhbyugfyj"

Right Boundary :"gytfytguihi

Variable name: token(name in which dynamic value to be store)

Regular expression extractor->abysiuhbyugfyj"abcd"gytfytguihi"efgh"ytdfuyguioio

Expression: abysiuhbyugfyj"(.+?)"gytfytguihi

Template : $1$

$1$ - (Capture value)abcd

Match No- (we can check total matches in view result tree after clicking reg exp extractor)

Expression: abysiuhbyugfyj"(.+?)"gytfytguihi"(.+?)"ytdfuyguioio

Template : $1$,$2$

$1$ - (Capture value)abcd

$2$ - (Capture value)efgh

Match No- (we can check total matches in view result tree after clicking reg exp extractor)

|  |  |
| --- | --- |
| . | matches any character |
| + or \* | any number of times |
| ? | stop from first match |

Diff (.\*?) and(.\*)?

* (.\*?) matches any character (.) any number of times (\*), as few times as possible to make the regex match (?).
* (.\*)? captures a group zero or one times (?). That group consists of a run of any length (\*) of any character (.)

Diff .+? and .\*?

.+? form requires at least one character to match,

.\*? can match none at all. Both operations are non-greedy, so they will try to find the shortest possible matching string.

22/09/2023

Task given based on the extractores

23/09/2023

Bit Bucker- repository to store our script

Http access token-access token- project admin-expiry(expire automatically -90days)

ensure no special character in expiry token - if u got regenerate it

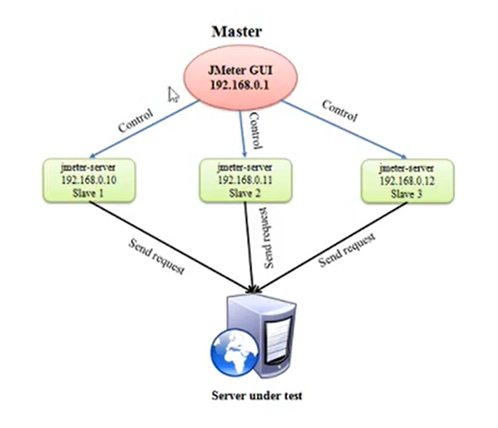
select the script you need to do and clone it

commends to clone : git clone <https://nissanXid:accestoken@bitbucket.aws.na.nissancloud.biz/scm/apx0375/service-schedule-perf-testing.git>

Jenkins - To execute our script in agent(Slave) machine(like performance centre)

Master slave mechanism(Distribution test ) - master and slave machine must have

* same Jmeter version
* same java version
* same network?



* Run-Suite :: This job handles the parallel execution of all the Jmeter script jobs configured on the Jenkins project space.
* Suite-Setup :: This job is used to setup the docker images required for the Jmeter executions on aws ECS platform

Suite-Setup-Pipeline

@Library('shared-library') \_

jmeterSuiteSetup {

appRegion = "NNA" // [MUST] App Region

projectId = "APXXXXX" // [MUST] APX ID or Project ID

GitURL = "<https://bitbucket.aws.na.nissancloud.biz/scm/XXXXX/repo_name.git>" // [MUST] Git URL for Performance testing repo.

GitBranch = "master" // [MUST] Git Branch for Performnce testing repo

awsTag= "" // [MUST] AWS Cost identification number for Project

}

Run-Suite-Pipeline

properties([

parameters([

string(defaultValue: "1", description: 'Number of Jmeter engines required. Default value is 1', name: 'JmeterEngineCount'),

string(description: 'Specify the Jmeter run time options here. eg: -Gthreads=100', name: 'jmeterOptions', defaultValue: "-Gthreads=1 -Gramptime=1 -Gduration=60 -Gdelay=1"),

])

])

@Library('shared-library') \_

jmeterRunSuite {

emailID = "" // [MUST] comma separated email id of end users, who requires the email notifications

}

Run-Script-Pipeline

properties([

parameters([

string(defaultValue: "1", description: 'Number of Jmeter engines required. Default value is 1', name: 'JmeterEngineCount'),

string(description: 'Specify the Jmeter run time options here. eg: -Gthreads=100', name: 'jmeterOpts1', defaultValue: "-Gthreads=1 -Gramptime=1 -Gduration=60 -Gdelay=1"),

])

])

@Library('shared-library') \_

jmeterRunScript {

appRegion = "NNA" // [MUST] App Region

projectId = "ApXXXX" // [MUST] ECS Service Name

GitURL = "<https://bitbucket.aws.na.nissancloud.biz/scm/XXXXX/repo_name.git>" // [MUST] Git URL for Performance testing repo.

GitBranch = "master" // [MUST] Git Branch for Performnce testing repo

awsTag= "Your\_AWS Cost tag here" // [MUST] AWS Cost identificatin bumber

emailID = "" // [MUST] comma separated email id of end users, who requires the email notifications

}

git cheat codes- to get all git comments.

BEANSHELL SCRIPTING

//ctx is a context class object, this class already exists in JMETER API

Ctx is used to fetch responses of the request

//Fetching Response Code

log.info("Response Code is + ctx.getPrevious Result().getResponseCode());

System.out.println("Response Code is + ctx.getPreviousResult().getResponseCode());

/Fetching Response Headers

log.info("Response Headers are + ctx.getPreviousResult().getResponseHeaders());

System.out.println("Response Headers are + ctx.getPreviousResult().getResponseHeaders());

/Fetching Request Body

log.info("Response Body is + ctx.getPreviousResult().getResponseDataAsString();

System.out.println("Response Code is + ctx.getPrevious Result().getResponseCode()); - did not match any documents.

System.out.println ->HELPS TO GET THE RESPONSE ON COMMAND PROMPT

**BEANSHELL ASSERTION**

if (ctx.getPreviousResult().getResponseCode().equals("400"))

{

Failure=false;

}

else

{

Failure=true; FailureMessage="stu us Code Assertion Failed, Expecting 4m"

}

**BeanShell Scripting - Fetching Different Data From Server Response**

log.info("Response Header Size is + ctx.getPrevious Result().getHeadersSize());

log.info("Response Body Size is " + ctx.getPrevious Result().getBodySizeAsLong());

log.info("Content Type is + ctx.getPreviousResult().getContentType());

log.info("Thread Name is + ctx.getPrevious Result().getThreadName());

log.info("Start Time is + ctx.getPrevious Result().getStartTime());

log.info("End Time is " + ctx.getPrevious Result().getEndTime());

log.info("Total Time is + ctx.getPreviousResult().getTime()); "

log.info("URL of the request is + ctx.getPrevious Result().getUrlAsString());

log.info("Error Count is + ctx.getPreviousResult().getErrorCount());

log.info("Latency is + ctx.getPreviousResult().getLatency());

BEANSHELL PREPROCCESOR - to capture URL from above request response and pass in the request before they execute

To add url

log.info("URL of the current request is + ctx.getVariables().get("URL"));

ctx.getVariables().put("URL","www.new Url");

To remove Url

ctx.getVariables().remove("URL");

To pass parameter value without CSV data set config - this pass single value for particular loop

 You have to put this request in loop controller to iterate it .

LOOP COUNT - 3

vars.put("URL","${\_StringFromFile(C:\Users\smore\Desktop\Testdata.txt)}");

|  |
| --- |
| Gokul.com |
| Raj.com |
| Mohan.com |

Loop.1-> URL = Gokul.com

Loop.2-> URL = Raj.com

Loop.3-> URL = Mohan.com

To pass all multiple value in the single request

They pass all data in single loop in variable name Content.

|  |
| --- |
| content |
| gokul |
| mohan |
| raj |
| dinesh |

To pass all these value in the single variable NAME

BufferedReader fileReader = new BufferedReader(new FileReader (vars.get("path")));

int counter = 1;

content = fileReader.readLine();

while((nextLine = fileReader.readLine()) != null)

{

content = content + "\n" + nextLine;

counter++;

}

vars.put("content", content);

x = vars.get("content");

**//Write data to file-File function -Write output in the external CSV file**

import java.io.\*;

File f=new File("C:\\Users\\smore\\Desktop\\TestResult.csv"); //C:\\Users\\smore\\Desktop\\TestResult.csv- file location where to write the test result.

FileWriter fw-new FileWriter (f); Bufferedwriter bw-new Bufferedwriter (fw);

var rc = prev.getResponseCode();

if (rc.equals("200"))

{

bw.write("Test Has Passed");

}

else{ bw.write("Test Has Failed");

}

bw.close();

fw close();