**SSW 567 - Assignment 10**

**Group 5**

Ed Chang

Harmony Sullivan

**Assignment Description**

Week 10: Performance & Reliability Testing

•Performance

You are to do a performance analysis of a system. You can try the ATM system, but i don't think it accepts multiple users. You need to have something which can have multiple users.

Step 1:

Create a simple workload model for the system for a typical hour on a typical day. I expect to see a number of users with different scenarios that then become test cases. What would you vary using user variables?

Step 2:

Try testing this with a performance load testing tool. Document what you did and analyze the results of the test. Did you find any performance problems?

There are lots of freeware tools. Pick one you think will run.

• Reliability

You are to estimate the reliability of this system. You will be using all dummy data/estimates since we don’t have any real data.

Create a small operational profile for this system (can use/adjust workload model from previous step.). Have between 5 and 10 operations /transactions only. If your system has lots of operations, you can either have very high-level operations or pick one smaller area to profile.

Now, using the data below (assume only 6 operations), calculate the system reliability.

|  |  |  |  |
| --- | --- | --- | --- |
| **Transaction/ Operation** | **Relative Freq.** | **Duration (secs)** | **Failure Rate per minute** |
| 1 | 50 | 10 | 0.01 |
| 2 | 20 | 15 | 0.02 |
| 3 | 10 | 18 | 0.03 |
| 4 | 2 | 22 | 0.04 |
| 5 | 5 | 6 | 0.05 |
| 6 | 80 | 10 | 0.06 |

* If you needed to improve the reliability, which transaction would you would on first?
* If the FIO (Failure Intensity Objective) was .02 per minute, is the system reliability good enough to release? Why or why not?
* Document your thoughts.

**Results**

***Performance***

**Step 1:**

***Workload Model for Online Shopping Company (OSC) for Pet Supplies***

* Up to 200 customers simultaneously performs transaction on the OSC site
* Average transactions lasts 2 minutes and 30 seconds (150 seconds)
* Up to total of 8000 transactions are taken place within an hour
* 68% of the transactions are to purchase merchandise
* 15% of the transactions are to return merchandise
* 10% of the transactions are to exchange merchandise
* 6% of the transactions are to inquire about merchandise – require additional assistance
* 1% of the transactions are administrative related – System/Site Administrator to apply changes/updates
* Every 60 days, the OSC site is scheduled to be offline for 10 minutes to perform maintenance
* Up to 3 System Administrators can simultaneous login in at a time

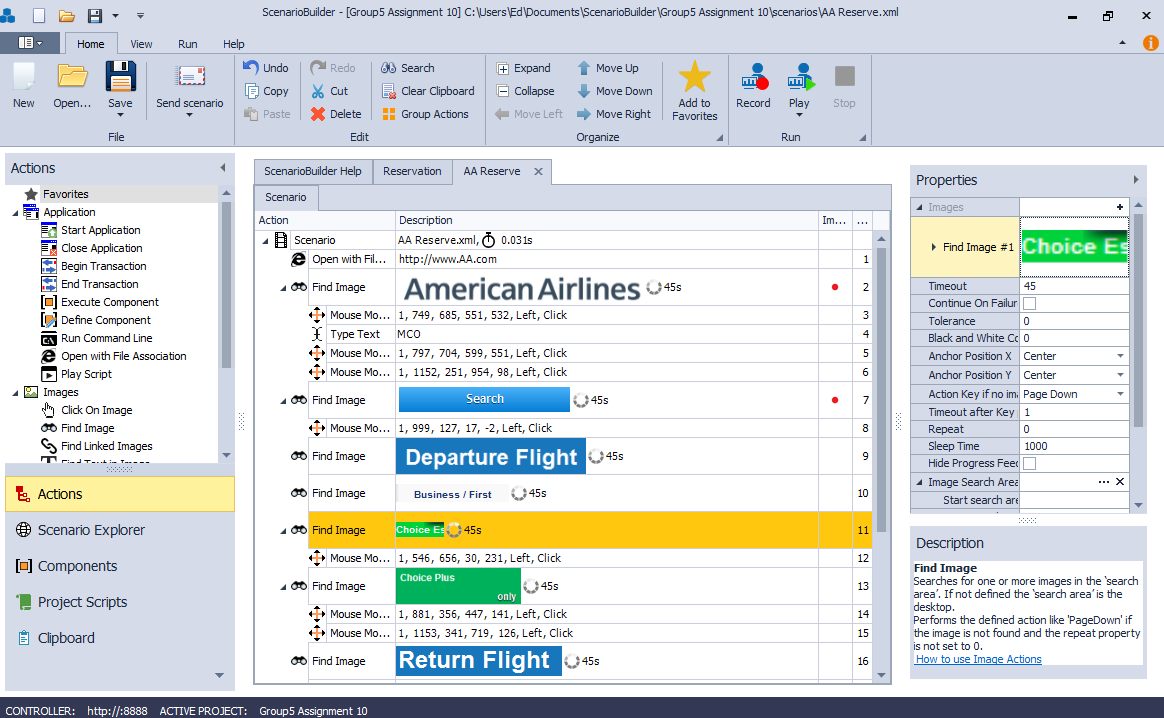
**Step 2:**

***Results from Load Testing Tools***

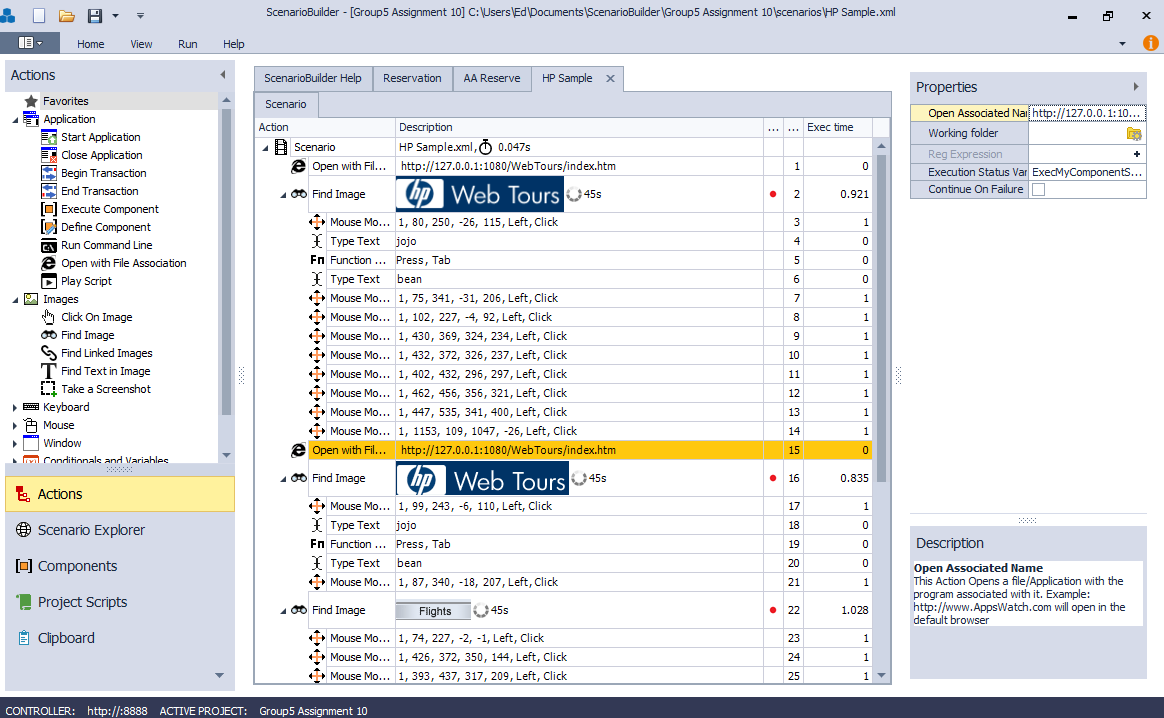
Various freeware tools were tried by the team, including HP’s LoadRunner, NRG Global’s AppLoader, SOASTA’s CloudTest Lite. As with other freeware that we have worked with throughout this semester, we ran into multiple issues while trying to use these tools. The biggest issue when we got started was the installation and configuration of these tools. After much trial-and-error, the best results we were able to captured was using AppLoader by NRG; however, even that was with plenty of setbacks.

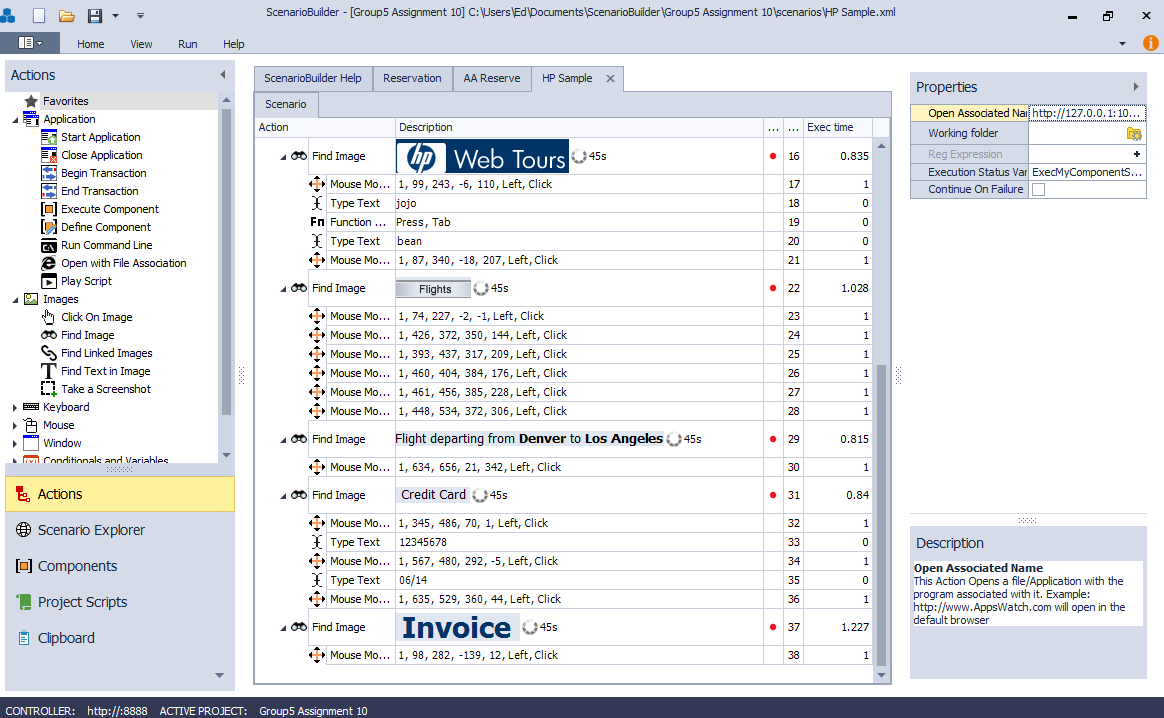
First of all, all these tools require a ‘real’ website in order to capture the interactions between the user and the website in order to create the test scenarios. Since most of the online shopping has dynamic loading of contents based on user preferences and user demands; we decided to use online airline purchasing as the test scenario instead – hoping the contents would be more stable than the online shopping sites.

However, we found out that even the online airline website had plenty of problems. Due to the dynamic status of the flights, airfares and information relevant to the customers, once a test scenario is created and captured, the playback of these test scenarios are inconsistent at best.

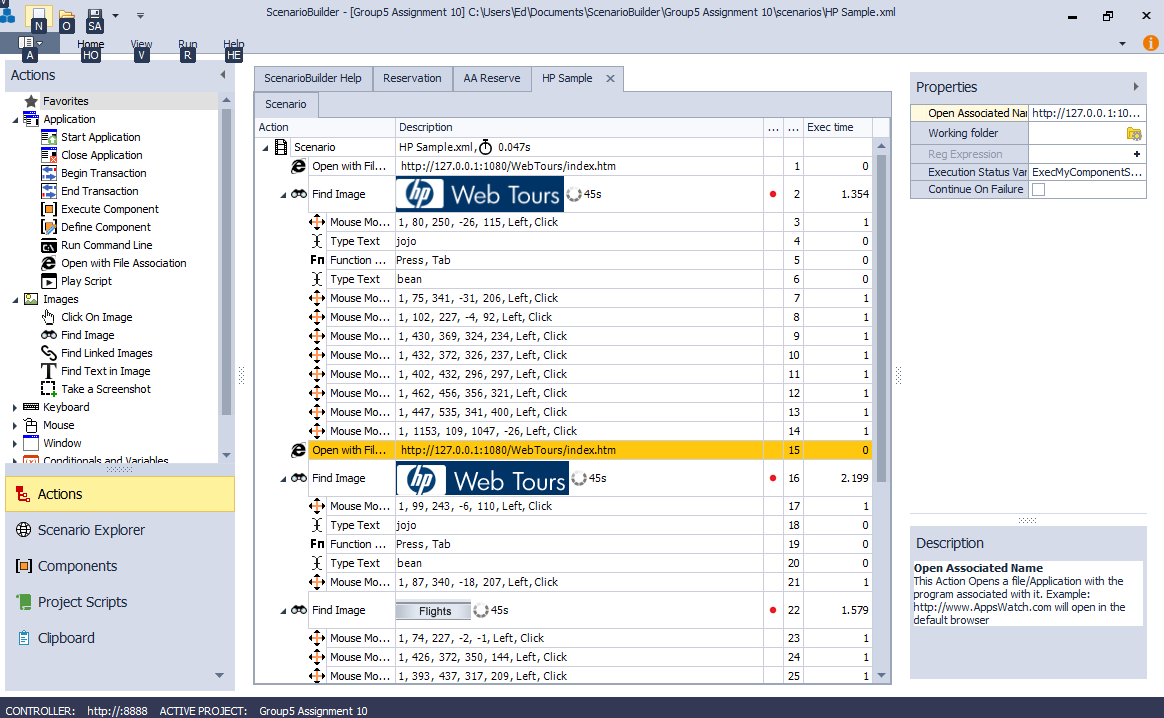


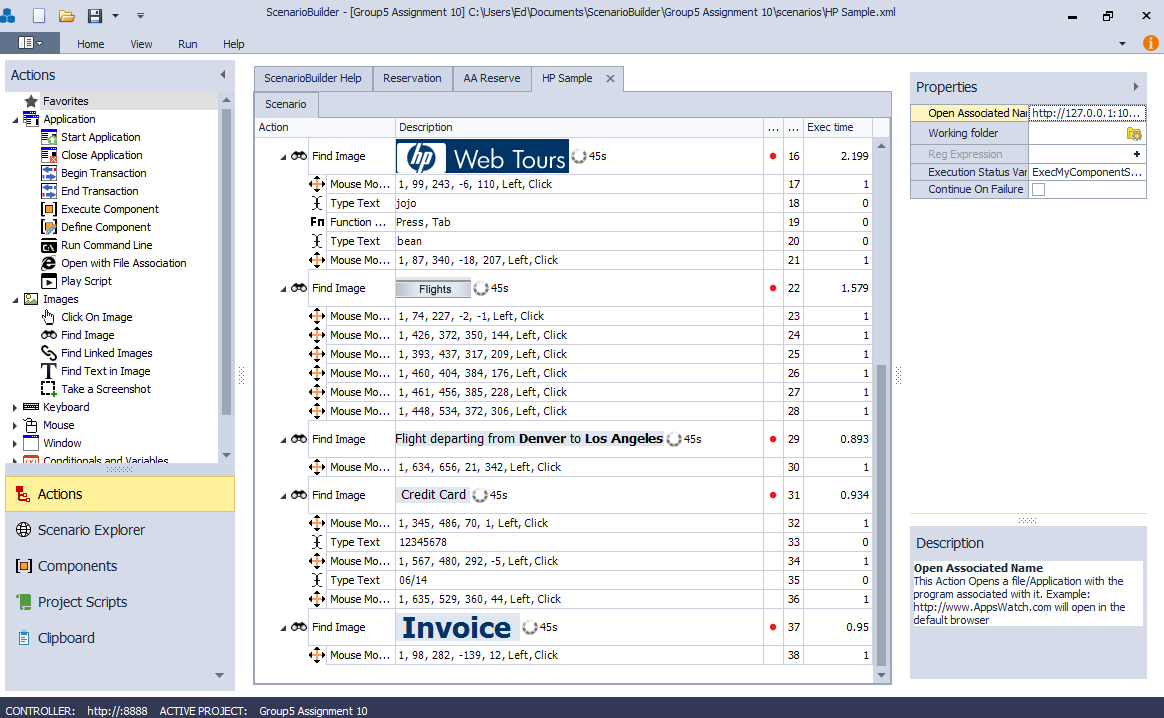
A compromise was made instead to use the sample Web App (Web Tours) from HP’s LoadRunner for the purchase ticket test scenario since it’s static and we can repeatedly conduct the test scenario instead constantly fine-tuning the test scenario based on the dynamic contents of the website.





Once the airline reservation test scenario was created, it was set to ‘Play Iterations’ in order to see what happens when the same test is repeated over time.





The results from this ‘Play Iterations’ indicated that the relevant steps takes longer to load/execute when they are repeated for multiple times. As shown in the table below, the same step took an average of 0.37 seconds longer to load/execute when repeatedly for 5 times. This indicates that this Web Tours application probably won’t scale well over time, i.e. optimization might be needed to fine tune in order to continue to support the customers over time.

|  |  |  |
| --- | --- | --- |
| **Step #** | **Single Run** | **Repeated Runs (5)** |
| 2 | 0.921 | 1.354 |
| 16 | 0.835 | 2.199 |
| 22 | 1.028 | 1.579 |
| 29 | 0.815 | 0.893 |
| 31 | 0.84 | 0.934 |
| 37 | 1.227 | 0.95 |
| **Total** | 5.666 | 7.909 |
| **Average** | 0.944 | 1.318 |

***Reliability***

***Operational Profile for OSC for Pet Supplies***

Based on the Performance part of this assignment, the following operational profile as well as the predicated aggregated reliability for the OSC system was constructed.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Transaction/ Operation** | **Relative Freq.** | **Relative Probability** | **Duration (secs)** | **Duation (min)** | **Duratin \* Freq** | **Operation Probability** | **Failure Rate per minute** | **Operation Contribution to Sy Failure Rate** |
| Purchase Item | 68 | 0.68 | 10 | 0.17 | 11.33 | 0.56 | 0.01 | 0.01 |
| Return Item | 15 | 0.15 | 15 | 0.25 | 3.75 | 0.18 | 0.02 | 0.00 |
| Exchange Item | 10 | 0.10 | 18 | 0.30 | 3.00 | 0.15 | 0.03 | 0.00 |
| Inquiry Item | 6 | 0.06 | 22 | 0.37 | 2.20 | 0.11 | 0.04 | 0.00 |
| Administrative | 1 | 0.01 | 6 | 0.10 | 0.10 | 0.00 | 0.05 | 0.00 |
|  |  |  |  |  |  |  |  |  |
| **Predicted Aggregate Reliability** | | |  |  |  |  |  | 0.02 |

***Reliability for Sample System***

The following table depicts the predicted aggregated reliability for the Sample System given.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Transaction/ Operation** | **Relative Freq.** | **Relative Probability** | **Duration (secs)** | **Duation (min)** | **Duratin \* Freq** | **Operation Probability** | **Failure Rate per minute** | **Operation Contribution to Sy Failure Rate** |
| 1 | 50 | 0.30 | 10 | 0.17 | 8.33 | 0.27 | 0.01 | 0.00 |
| 2 | 20 | 0.12 | 15 | 0.25 | 5.00 | 0.16 | 0.02 | 0.00 |
| 3 | 10 | 0.06 | 18 | 0.30 | 3.00 | 0.10 | 0.03 | 0.00 |
| 4 | 2 | 0.01 | 22 | 0.37 | 0.73 | 0.02 | 0.04 | 0.00 |
| 5 | 5 | 0.03 | 6 | 0.10 | 0.50 | 0.02 | 0.05 | 0.00 |
| 6 | 80 | 0.48 | 10 | 0.17 | 13.33 | 0.43 | 0.06 | 0.03 |
|  |  |  |  |  |  |  |  |  |
| **Predicted Aggregate Reliability** | | |  |  |  |  |  | 0.04 |

If we were to improve the reliability of this Sample System, i.e. reduce the failure rate, the transaction / operation that we would recommend to improve first would be ‘Transaction/Operation’ #6 since it’s the major contributor to the overall System Failure Rate, e.g. 75% (0.03/0.04) of so of failures were caused by this transaction / operation alone.

When a system has a Failure Intensity Rate (FIO) of 0.02 per minute, whether or not to release this system is depend upon what is the nature of the given system. For something that is safety critical, i.e. brake system on an aircraft or train, 0.02 failures per minute is equate to 28.8 failures per day; this would not be tolerable since many lives are in stake. However, the same FIO would be perfectly fine for a Printer since that’s in stake is only the users resubmit the print job or finding a different printer to use.

**Lessons Learned**

Similar to other assignments during this semester, the most challenging part of the assignment has been trying to find the right tool for our environment, setting up the tool and using the tool to create the scenarios stated in the assignment.

In order to complete the Load Testing part of the assignment, we had to use parts of two different tools to complete that: 1) to mimic the operational environment, i.e. the website, and 2) to capture and to conduct test scenarios. As part of that, the original system (online shopping) we had envision had to be abandoned during this part of the testing since our system did not match with the ‘operational environment’ that was available for us to conduct our testing.

With proper planning and resource availability, a project should start planning for the Load and Reliability Testing early in the project phase. That way, the tool suite can be identified, familiarized and utilized prior to the need date; thus there isn’t a last minute ramp-up to setup the testing environment. Or worse, bypass this part of the testing as a whole.

In addition, due to the dynamic nature of the web contents presented to the users nowadays, it’s important to ensure testing as this is done in a sandbox so the variables are reduced to a minimum. We found out that when we tried to create test scenarios from the American Airlines website that too many variables (cookie settings, user profiles) could cause the website to render differently thus causing the repeating test scenarios to fail. Without any control over these variables, it’s hard to judge how well the system is going to perform under these conditions.

**Honor Pledge**

We pledge on our honor that we have not given or received any unauthorized assistance on this assignment/examination. We further pledge that we have not copied any material from a book, article, the Internet or any other source except where I have expressly cited the source.