CHAPTER 5

TESTING

In this section we are presenting the various testing responses which have been performed on PReWebD and PReWebN by deploying the two applications in Mercury LoadRunner 8.0. We also present the results of different statistical tests which have been performed on the data recorded during various testings.

5.1 The Performance Testing

The performance testing is a test which is performed, from one perspective, to determine how fast some aspect of a system performs under a particular workload. It can also serve to validate and verify other quality attributes of the system, such as scalability, reliability and resource usage. It can also demonstrate whether the system meets performance criteria or not. It can compare two systems to find out which one performs better. It can also diagnose the part of software that contributes most to the poor performance of the system [73].

After verifying the correctness of the code, the load and stress testing is performed on PReWebD and PReWebN to measure the performance and scalability of the application under heavy load. After analyzing the results obtained during this phase, it become possible for us to find out the bottlenecks, memory leakage and performance problems related to the database layer.

The load testing is performed on PReWebD and PReWebN to have an overall insight of both the systems. It models the behavior of users in real world. The load generator mimics browser behavior. Each emulated browser is called a virtual user, which is the main load testing concept. A load test is valid only if virtual users' behavior has characteristics similar to those of actual users [74]. In load testing, the application is subjected through reasonable load in terms of number of virtual users to find out the performance of the system, mainly in terms of response time. In this case the load is varied from zero to the maximum up to which the application can handle in a decent manner.
The stress testing is performed to determine the stability of PReWebD and PReWebN. It involves testing beyond the normal operational capacity. In this process the PReWebD and PReWebN is subjected to unreasonable load. The stress testing is performed to uncover memory leakage, bandwidth limits, transactional problems, resource locking, hardware limitations and synchronization problems that may occur when the applications are loaded beyond the limits determined by the performance statistics [75, 76]. The motive is to determine the break point at the weakest link within the entire system and to observe whether the PReWebD and PReWebN recover gracefully or not. The responses are then observed to find out the bugs that will make the applications potentially harmful [77].

5.2 The Performance Testing Approach for PReWebD and PReWebN

The different steps involved in performing performance test on our prototype PReWebD and PReWebN are discussed below:

(i) The preparation of test plan

Various test cases are designed to view performance test from user’s perspective. The test cases include the stress levels for each performance indicators, number of users, architectures which include client PC, browsers, network bandwidth etc.

(ii) The creation of test environment

The test environment consists of hardware and software configuration of client PC, operating system chosen, and number of client PC participating in the test etc.

(iii) The setting of test period

The period for which the stress test will be performed is decided in this step. There may be three phases: (a) the ramp up phase initializes the system until it reaches a steady state, (b) the steady state phase where the measurements are taken and (c) the ramp down phase which allows the PReWebD and PReWebN to cool down.
(iv) The setting of performance parameters

The performance parameters are important metrics to measure the performance of the PReWebD and PReWebN. All the parameters are not relevant to us. The parameters which are important to our present work are set, e.g. response time, throughput, hits/sec, errors/sec etc.

(v) The execution of the test

After setting the performance parameters the test is executed by deploying the applications on Mercury LoadRunner.

(vi) The analysis of test results

The test responses which are in forms of graphs are then analyzed to evaluate the performance of PReWebD and PReWebN.

The flow chart of the test plan for PReWebD and PReWebN is shown in fig.5.1.

5.3 The Testing Parameters

There are three main parameters which are varied during the testing procedure. They are: (a) the workload intensity measured in terms of number of virtual users, i.e. stress level, (b) the workload mix which will define what users will do in each session and (c) the user behavior parameter, which is the think time.

5.4 The Test Responses

The parameters of the load and stress test which we have monitored include: (i) the response time in terms of second, (ii) the throughput in bytes per second, (iii) the hits per second, (iv) the number of successful virtual users allowed for transaction, (v) the transaction summary which includes the number of completed and abandoned sessions, (vi) the error report etc.

The fig.5.2(a) and fig.5.2(b) shows a sample screen shot for testing PReWebD as well as PReWebN using Mercury LoadRunner.
Fig.5.1: The Flow Chart for test plan of PReWebD and PReWebN
Fig.5.2(a): Screen shot for testing using Mercury LoadRunner for PReWebD

Fig.5.2(b): Screen shot for testing using Mercury LoadRunner PReWebN
5.5 The Performance Testing and Evaluation of PReWebD

The performance testing is performed on PReWebD for 10, 20, 30, 40, 50, 75, 100 and 125 virtual users. All parameters are measured at 128 kbps bandwidth. We observed various responses provided by the LoadRunner. All the performance tests are conducted with ramp up schedule of 1 virtual user operating in every 30 seconds. The steady state measurement period is set at 30 minutes duration. Then they are phased out at the same time after the completion of the steady state period. The delays for the users think time is included to emulate the behavior of real users. On observing the responses, it is seen that up to 40 users the application runs smoothly. The results of testing for various operations are discussed below.

5.5.1 The Insert Operation

The various responses for testing of PReWebD on insert operation are discussed below:

5.5.1.1 The Stress Level vs. Number of Hits

The fig. 5.3 shows the plot for hits/sec vs. number users for 75 virtual users. In this case 69 users are allowed to perform the transaction, rest are failed. It is observed that hits/sec increases with the number of virtual users. It becomes maximum at around 63 virtual users and then the parameter decreases gradually. The recorded average hits/sec is 12.241 with maximum of 19.539.

The fig. 5.4 shows the plot for hits/sec vs. number of users for 100 virtual users. In this case only 65 users are allowed for transaction, rests are failed. Here also, the number of hits increases with the number of virtual users, becomes maximum at 63 and decreases gradually. The recorded average hits/sec is 8.963 with a maximum of 18.754.

The fig. 5.5 shows the plot for hits/sec vs. number of users for 125 virtual users. In this case only 20 users are allowed for transaction and rests are failed. Here also, the number of hits increases with the number of virtual users, becomes maximum at 10 and decreases gradually. The recorded average hits/sec is 1.515 with a maximum of 4.4.
5.5.1.2 The Stress Level vs. Throughput

The fig.5.6 shows the plot for throughput vs. number of users for 75 virtual users. It is observed that the throughput increases with the number of virtual users. It becomes maximum at around 63 virtual users and then the parameter decreases gradually. The recorded average throughput is 328940.716 with maximum of 607717.333 bytes per second.

The fig.5.7 shows the plot for throughput vs. number of users for 100 virtual users. Here also, the throughput increases with the number of virtual users, becomes maximum at 63 and decreases gradually. The recorded average throughput is 308799.512 with a maximum of 602554.6 bytes per second.

The fig.5.8 shows the plot for throughput vs. number of virtual users for 125 users. Here also, the throughput increases with the number of virtual users, becomes maximum at 13 and decreases gradually. The recorded average throughput is 52673.631 with a maximum of 77509.4.

5.5.1.3 The Stress Level vs. Response Time

The fig.5.9 shows the plot for response time vs. number of users for 75 virtual users. It is observed that the response time increases initially with the number of virtual users, then it reached a steady state and then increases to a maximum level with increasing virtual users. The recorded average response time is 45.86s with maximum of 57.801s.

The fig.5.10 shows the plot for response time vs. number of users for 100 virtual users. It is observed that the response time increases initially with the number of virtual users, then it reached a steady state and then increases to a maximum level for 30 to 35 users and decreases gradually to a steady state. The recorded average response time is 49.859s with maximum of 158.145s.

The fig.5.11 shows the plot for response time vs. number of users for 125 virtual users. The recorded average response time is 76.646s with maximum of 159.391s.
Fig. 5.3: Hits/sec vs. number of users for 75 virtual users for insert operation

Fig. 5.4: Hits/sec vs. number of users for 100 virtual users for insert operation
Fig.5.5: Hits/sec vs. number of users for 125 virtual users for insert operation

Fig.5.6: Throughput vs. number of users for 75 virtual users for insert operation
Fig. 5.7: Throughput vs. number of users for 100 virtual users for insert operation

Fig. 5.8: Throughput vs. number of users for 125 virtual users for insert operation
Fig. 5.9: Response time vs. number of users for 75 virtual users for insert operation

Fig. 5.10: Response time vs. number of users for 100 virtual users for insert operation
Fig. 5.11: Response time vs. number of users for 125 virtual users for insert operation
5.5.2 The Delete Operation

The testing on delete operation is performed and monitored for 40 virtual users. In this case all the virtual users were allowed to perform the transactions. Various output responses of testing on delete operation are discussed below:

5.5.2.1 The Response Time

The fig.5.12 shows the plot for response time vs. number of users in case of delete operation. The response time increases from the minimum value, rises to maximum at 12 to 17 users and falls to maintain the average value. The recorded average response time is 87.17 sec.

5.5.2.2 The Throughput

The fig.5.13 shows the plot for throughput vs. number of users. In this case, the throughput increases with the number of virtual users to attain a maximum value, then drops to a lower value between 33 to 36 users. Again it rises to attain high throughputs. The average throughput is 227313.979 bytes/sec.

5.5.2.3 The Hits/sec

The fig.5.14 shows the plot for hits/sec vs. number of users. Here, hits/sec increases with the increasing number of virtual users. The average value is 3.451.
Fig. 5.12: Response time vs. number of users for 40 virtual users for delete operation

Fig. 5.13: Throughput vs. number of users for 40 virtual users for delete operation
Fig. 5.14: Hits/sec vs. number of users for 40 virtual users for delete operation
5.6 Statistical Testing for PReWebD

The statistical analysis for 10 users run for 5 minutes in steady state is carried out on observed data for PReWebD. The replica of the test response has been recorded. The observed data are given in Table 5.1 to analyze for evaluation of the performance of the application.

<table>
<thead>
<tr>
<th>No. of Observations</th>
<th>Response Time (sec)</th>
<th>Hits/sec</th>
<th>Throughput (bytes/sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>47.244</td>
<td>1.461</td>
<td>60090.519</td>
</tr>
<tr>
<td>2</td>
<td>47.307</td>
<td>1.637</td>
<td>61701.468</td>
</tr>
<tr>
<td>3</td>
<td>52.713</td>
<td>1.739</td>
<td>63793</td>
</tr>
<tr>
<td>4</td>
<td>48.174</td>
<td>1.634</td>
<td>58633</td>
</tr>
<tr>
<td>5</td>
<td>51.512</td>
<td>1.857</td>
<td>62692</td>
</tr>
<tr>
<td>6</td>
<td>50.384</td>
<td>1.81</td>
<td>63637</td>
</tr>
<tr>
<td>7</td>
<td>50.156</td>
<td>1.773</td>
<td>63346</td>
</tr>
<tr>
<td>8</td>
<td>51.505</td>
<td>1.8</td>
<td>64009</td>
</tr>
<tr>
<td>9</td>
<td>51.212</td>
<td>1.74</td>
<td>61049</td>
</tr>
<tr>
<td>10</td>
<td>58.867</td>
<td>1.831</td>
<td>65033</td>
</tr>
<tr>
<td>11</td>
<td>60.425</td>
<td>1.921</td>
<td>63128</td>
</tr>
<tr>
<td>12</td>
<td>58.414</td>
<td>1.8</td>
<td>66928</td>
</tr>
<tr>
<td>13</td>
<td>57.137</td>
<td>1.8</td>
<td>63315</td>
</tr>
<tr>
<td>14</td>
<td>56.105</td>
<td>1.738</td>
<td>62289</td>
</tr>
<tr>
<td>15</td>
<td>58.556</td>
<td>1.87</td>
<td>65492</td>
</tr>
<tr>
<td>16</td>
<td>51.63</td>
<td>1.664</td>
<td>60312</td>
</tr>
<tr>
<td>17</td>
<td>49.763</td>
<td>1.68</td>
<td>60061</td>
</tr>
<tr>
<td>18</td>
<td>49.859</td>
<td>1.7</td>
<td>61927</td>
</tr>
<tr>
<td>19</td>
<td>53.627</td>
<td>1.71</td>
<td>63816</td>
</tr>
<tr>
<td>20</td>
<td>54.118</td>
<td>2.06</td>
<td>68755</td>
</tr>
<tr>
<td>21</td>
<td>52.816</td>
<td>1.76</td>
<td>64853</td>
</tr>
<tr>
<td>22</td>
<td>51.734</td>
<td>1.395</td>
<td>62305</td>
</tr>
<tr>
<td>23</td>
<td>58.028</td>
<td>1.81</td>
<td>65107</td>
</tr>
<tr>
<td>24</td>
<td>60.194</td>
<td>1.795</td>
<td>64035.426</td>
</tr>
<tr>
<td>25</td>
<td>56.779</td>
<td>1.558</td>
<td>63157</td>
</tr>
<tr>
<td>26</td>
<td>54.177</td>
<td>1.553</td>
<td>61662.246</td>
</tr>
<tr>
<td>27</td>
<td>58.143</td>
<td>1.748</td>
<td>64080.816</td>
</tr>
<tr>
<td>28</td>
<td>57.065</td>
<td>1.761</td>
<td>62954.72</td>
</tr>
<tr>
<td>29</td>
<td>53.163</td>
<td>1.8</td>
<td>64437.829</td>
</tr>
<tr>
<td>30</td>
<td>57.204</td>
<td>1.681</td>
<td>60205.179</td>
</tr>
</tbody>
</table>

5.6.1 The Normality Test for PReWebD

Our objective is to determine the distribution of the response time, hits/sec and throughputs. The easiest way of determination is to plot histograms of the observed
response times, hits/sec and throughputs as shown in fig.5.15, fig.5.16 and fig.5.17 respectively. But there is a major problem with histograms, that is, depending on the used bin size it is possible to draw very different conclusions. A better technique is to plot the observed quantiles versus the recorded data in a quantile plot [24].

The applied distribution is normal distribution according to the histograms. Based on the observation of histograms in fig.5.15, fig.5.16, fig.5.17, the response times do appear to be normally distributed. The normality test can also be done by using quantile plots. If the data samples are taken from a normal distribution, the plot will appear to be linear. The quantile plots of the response time, throughput and hits/sec are shown in fig.5.18, fig.5.19 and fig.5.20 respectively. The data follows a straight line which predicts that the distribution is a normal one [78].

The normality test can be verified graphically using the normal probability plot. If the data samples are taken from a normal distribution, the plot will appear to be linear. The normal probability plot of the response time, throughput and hits/sec are shown in fig.5.21, fig.5.22 and fig.5.23 respectively. The data follows a straight line which predicts that the distribution is a normal one. But sometimes the data departs from the line at ends. This implies that the data may have longer tails than the normal distribution.

5.6.2 The Confidence interval of response time, hits/sec and throughput

The 95% confidence interval for the mean values of response time, hits/sec and throughputs are estimated from the observed data. The population means $\mu$ can be expressed as [79, 80].

$$\mu = \bar{x} \pm \frac{t_c S}{\sqrt{N}}$$

(1.1)

Where $\bar{x}$ = mean value, $t_{c(0.05, 29)}$ = critical value, $S$ = standard deviation, $N$ = sample size and $\frac{t_c S}{\sqrt{N}}$ = margin of errors.

Considering different values for parameters, obtained during load testing, we evaluate the critical value, mean, and margin of errors which are given in Table 5.2. The population mean $\mu$ is calculated from equation (1.1).
Fig. 5.15: The Histogram for response time

Fig. 5.16: The Histogram for hits/sec
Fig. 5.17: The Histogram for throughput

Fig. 5.18: The Quantile plot for response time
Fig. 5.19: The Quantile plot for hits/sec

Fig. 5.20: The Quantile plot for throughput
Fig. 5.21: The Normal probability plot for response time

Fig. 5.22: The Normal probability plot for hits/sec
Fig. 5.23: The Normal probability plot for throughput
From Table 5.2, we observed that (i) the mean response time lies between $53.9337 \pm 1.455$ i.e. 52.478 to 55.389 sec, (ii) the mean hits/sec lies between 1.686 and 1.785 and (iii) the mean throughput lies between 62289.773 to 63898.226.

### 5.6.3 The Factors Influencing the Response Time

In order to verify whether a relationship between response time, hits/sec and throughput exists, we assume that such a relation, if exists, be a linear one. The response time is assumed as criterion variable. Hits/sec and throughput are assumed to be predictor variables. The scatter plots of observed response time vs. hit/sec and throughput are shown in fig.5.24 and fig.5.25 respectively.

The two scatter plots with their respective regression line shows linear relationship. Greater the value of hits/sec, more will be the response time, however very small. Similarly, the greater the value of throughput more will be the response time.

To examine the combined effect of throughput and hits/sec on response time, we performed multiple linear regression test. The test is carried out at 95% confidence level. We assumed the null hypothesis ($H_0$) which is: response time does not depend on hits/sec and throughput. The alternate hypothesis ($H_1$) is: response time is a function of hits/sec and throughput.

The regression analysis calculates F ratio to be 5.087 which is significant at 0.05. This implies that there exists a linear relationship between response time, hits/sec, and throughput. As such, the null hypothesis may be rejected. This implies that the equation has 95% chance of being true. The analysis also suggested that our model accounts for 27.37% variance on response time. Thus we may infer that the hits/sec and throughput have some influence on response time. The linear regression equation evaluated from our experimental results is of the form given in equation (1.2)

$$\text{Response time} = -0.0467 + 4.584 \times \text{hits/sec} + 0.000736 \times \text{throughput} + \text{residue} \quad (1.2)$$
Fig. 5.24: The scatter plot of Hits/sec vs. Response time

\[ y = 0.0009x - 4.3233 \]
\[ R^2 = 0.2602 \]

Fig. 5.25: The scatter plot of Throughput vs. Response time

\[ y = 12.596x + 32.064 \]
\[ R^2 = 0.1818 \]
5.7 The Performance Testing and Evaluation of PReWebN

The performance testing similar to that carried out on PReWebD is also performed on PReWebN for 10, 20, 30, 40, 50, 75, 100 and 125 virtual users for insert and delete operations. All parameters are measured at 128 kbps bandwidth. Here also the performance tests are conducted with ramp up schedule with 1 virtual user operating every 30 seconds. The steady state measurement period is set at 30 minutes duration. Then they are phased out at the same time after the completion of the steady state period. The delays for the users think time is included to emulate the behavior of real users [81]. We observed various response parameters. It is seen that up to 50 users the application runs smoothly. The results of testing for various operations are discussed below.

5.7.1 The Insert Operation

The various responses for testing of PReWebN on insert operation are discussed below:

5.7.1.1 The Stress Level vs. Hits/sec

The fig.5.26 shows the plot for hits/sec vs. number of users for 75 virtual users. In this case 56 users are allowed to perform the transaction, rest are failed. It is observed that hits/sec increases with number of virtual users, becomes maximum at around 30 virtual users and then decreases gradually. The recorded average hits/sec is 3.681 with maximum of 10.333.

The fig.5.27 shows the plot for hits/sec vs. number of users for 100 virtual users. In this case only 49 users are allowed for transactions and rests are failed. Here also, the number of hits increases with number of virtual users, becomes maximum at 42 and decreases gradually. The recorded average hits/sec is 2.524 with maximum of 4.

The fig.5.28 shows the plot for hits/sec vs. number of users for 125 virtual users. In this case only 47 users are allowed for transactions, rests are failed. Here also, the number of hits increases with number of virtual users, becomes maximum at 27 and decreases gradually. The recorded average hits/sec is 2.6 with a maximum of 6.8.
The fig.5.29 shows the plot for throughput vs. number of users for 75 virtual users. It is observed that the throughput increases with the number of virtual users, becomes maximum at around 50 virtual users and then decreases gradually. The recorded average throughput is 129440.023 bytes per second with a maximum of 269202.1.

The fig.5.30 shows the plot for throughput vs. number of users for 100 virtual users. Here also, the throughput increases with the number of virtual users, becomes maximum at 38 and then decreases gradually. The recorded average throughput is 122645.04 with a maximum of 295156.3 bytes per second.

The fig.5.31 shows the plot for throughput vs. number of users for 125 virtual users. Here also, the throughput increases with the number of virtual users, becomes maximum at 45 and then decreases gradually. The recorded average throughput is 103503.55 bytes per sec.

The fig.5.32 shows the plot for response time vs. number of users for 75 virtual users. It is observed that the response time increases initially with number of virtual users, then it reached a steady state and then shoots to a maximum level with increasing virtual users at around 51 users. The recorded average response time is 124.876s with maximum of 251.156s.

The fig.5.33 shows the plot for response time vs. number of users for 100 virtual users. It is observed that the response time increases initially with number of virtual users, then increases abruptly from 25 users. The recorded average response time is 139.412s with maximum of 255.088s.

The fig.5.34 shows the plot for response time vs. number of users for 125 virtual users. It rises steadily up to 35 users and then increases abruptly. The recorded average response time is 129.12s with maximum of 267.766s.
Fig. 5.26: Hits/sec vs. number of users for 75 virtual users for insert operation

Fig. 5.27: Hits/sec vs. number of users for 100 virtual users for insert operation
Fig. 5.28: Hits/sec vs. no. of users for 125 virtual users for insert operation

Fig. 5.29: Throughput vs. number of users for 75 virtual users for insert operation
Fig. 5.30: Throughput vs. number of users for 100 virtual users for insert operation

Fig. 5.31: Throughput vs. no. of users for 125 virtual users for insert operation
Fig. 5.32: Response time vs. number of users for 75 virtual users for insert operation

Fig. 5.33: Response Time vs. number of users for 100 virtual users for insert operation
Fig. 5.34: Response time vs. no. of users for 125 virtual users for insert operation
5.7.2 The Delete Operation

The testing on delete operation is performed and monitored for 40 virtual users. In this case all the virtual users are allowed to perform the transaction. Various output responses are discussed below:

5.7.2.1 The Response Time

The fig.5.35 shows the plot for response time vs. number of users. The response time remains constant all throughout the test with an average value of 119.678 sec.

5.7.2.2 The Throughput

The fig.5.36 shows the plot for throughput vs. number of users. In this case, throughput increases with the increase in number of virtual users. The average throughput is 255967.868 sec.

5.7.2.3 The Hits/sec

The fig.5.37 shows the plot for hits/sec vs. number of users. In this case, hits/sec increases with the increase in number of virtual users. The average hits/sec is 3.042.
Fig. 5.35: Response time vs. number of users for 40 virtual users for delete operation

Fig. 5.36: Throughput vs. number of users for 40 virtual users for delete operation
Fig. 5.37: Hits/sec vs. number of users for 40 virtual users for delete operation
5.8 Statistical testing for PReWebN

The statistical analysis for 10 users run for 5 minutes in steady state is carried out for observed data on PReWebN. The replica of the test response has been recorded. The observed data are given in Table 5.3 to analyze for evaluation of the performance of the application.

Table 5.3: The observed data

<table>
<thead>
<tr>
<th>No. of observations</th>
<th>Response Time (sec)</th>
<th>Hits/sec</th>
<th>Throughput (bytes/sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>126.294</td>
<td>1.003</td>
<td>60552</td>
</tr>
<tr>
<td>2</td>
<td>127.143</td>
<td>1.067</td>
<td>61051</td>
</tr>
<tr>
<td>3</td>
<td>128.416</td>
<td>1.069</td>
<td>61736</td>
</tr>
<tr>
<td>4</td>
<td>129.311</td>
<td>1.053</td>
<td>62216</td>
</tr>
<tr>
<td>5</td>
<td>130.268</td>
<td>1.063</td>
<td>62883</td>
</tr>
<tr>
<td>6</td>
<td>131.325</td>
<td>1.021</td>
<td>63242</td>
</tr>
<tr>
<td>7</td>
<td>130.316</td>
<td>1.046</td>
<td>63474</td>
</tr>
<tr>
<td>8</td>
<td>134.448</td>
<td>1.04</td>
<td>64123</td>
</tr>
<tr>
<td>9</td>
<td>134.077</td>
<td>1.027</td>
<td>64377</td>
</tr>
<tr>
<td>10</td>
<td>135.119</td>
<td>1.067</td>
<td>65014</td>
</tr>
<tr>
<td>11</td>
<td>133.211</td>
<td>1.063</td>
<td>65447</td>
</tr>
<tr>
<td>12</td>
<td>137.164</td>
<td>1.081</td>
<td>65873</td>
</tr>
<tr>
<td>13</td>
<td>138.231</td>
<td>1.086</td>
<td>66486</td>
</tr>
<tr>
<td>14</td>
<td>120.684</td>
<td>1.063</td>
<td>62761</td>
</tr>
<tr>
<td>15</td>
<td>123.458</td>
<td>1.056</td>
<td>63230</td>
</tr>
<tr>
<td>16</td>
<td>124.26</td>
<td>1.049</td>
<td>63694</td>
</tr>
<tr>
<td>17</td>
<td>123.721</td>
<td>1.04</td>
<td>64858</td>
</tr>
<tr>
<td>18</td>
<td>124.888</td>
<td>1.021</td>
<td>65155</td>
</tr>
<tr>
<td>19</td>
<td>127.18</td>
<td>1.012</td>
<td>62258</td>
</tr>
<tr>
<td>20</td>
<td>119.8</td>
<td>1</td>
<td>61207</td>
</tr>
<tr>
<td>21</td>
<td>131.9</td>
<td>1.067</td>
<td>63998</td>
</tr>
<tr>
<td>22</td>
<td>128.457</td>
<td>1.056</td>
<td>62938</td>
</tr>
<tr>
<td>23</td>
<td>120.731</td>
<td>1.046</td>
<td>63155</td>
</tr>
<tr>
<td>24</td>
<td>118.269</td>
<td>1.04</td>
<td>64646</td>
</tr>
<tr>
<td>25</td>
<td>129.62</td>
<td>1.049</td>
<td>68837</td>
</tr>
<tr>
<td>26</td>
<td>130.9</td>
<td>1.075</td>
<td>66667</td>
</tr>
<tr>
<td>27</td>
<td>123.672</td>
<td>1.053</td>
<td>67484</td>
</tr>
<tr>
<td>28</td>
<td>135.233</td>
<td>1.09</td>
<td>68028</td>
</tr>
<tr>
<td>29</td>
<td>132.411</td>
<td>1.08</td>
<td>69308</td>
</tr>
<tr>
<td>30</td>
<td>133.067</td>
<td>1.085</td>
<td>67600</td>
</tr>
</tbody>
</table>
5.8.1 The Normality Test for PReWebN

As in the case of PReWebD, here also our objective is to determine the distribution of response time, hits/sec and throughputs. The procedure adopted is similar to that followed in case of PReWebD. The plots of the histograms for the observed parameters as in Table.5.3 are shown in fig.5.38, fig.5.39 and fig.5.40 respectively. The applied distribution is normal distribution according to the histograms.

The quantile plots of observed data for response time, throughput and hits/sec are plotted and shown in fig.5.41, fig.5.42 and fig.5.43 respectively. Based on the observed plots, the response time, throughput and hits/sec do appear to be normally distributed.

The test of the normality is verified graphically using the normal probability plot as done in the case of PReWebD. The normal probability plots of the response time, throughput and hits/sec are shown in fig.5.44, fig.5.45 and fig.5.46 respectively. These data also follows a straight line, which shows that the distribution is a normal one.

However it is observed that the distributions are not perfectly linear. It indicates that the data has longer tails than normal distribution.

5.8.2 The Confidence Interval of Response Time, Hits/sec and Throughput

The 95% confidence interval for the mean values of response time, hits/sec and throughputs are also estimated on observed data for PReWebN.

Considering the parameters, obtained during load testing, we evaluate the critical value, mean, and margin of errors for PReWebN which are shown in Table.5.4.

The evaluation of population mean \( \mu \) is calculated from equation (1.1)

<table>
<thead>
<tr>
<th>N</th>
<th>( t_{0.05, 29} )</th>
<th>Parameters</th>
<th>( \bar{x} )</th>
<th>S</th>
<th>( \frac{t_{p}S}{\sqrt{N}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>2.045</td>
<td>Response Time (sec)</td>
<td>128.7858</td>
<td>5.34267</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Hits/sec</td>
<td>1.052</td>
<td>0.024</td>
<td>0.009</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Throughput (bytes/sec)</td>
<td>64410</td>
<td>2322</td>
<td>867</td>
</tr>
</tbody>
</table>
Fig. 5.38: The Histogram for response time

Fig. 5.39: The Histogram for hits/sec
Fig. 5.40: The Histogram for throughput

Fig. 5.41: The Quantile plot for response time
Fig. 5.42: The Quantile plot for hits/sec

Fig. 5.43: The Quantile plot for throughput
Fig. 5.44: The Normal probability plot for response time

Fig. 5.45: The Normal probability plot for hits/sec
Fig. 5.46: The Normal probability plot for throughput
From Table 5.4, we observe that (a) the mean response time lies between 128.7858 ± 2 i.e. 126.8 to 130.78 sec, (b) the mean hits/sec lies between 1.043 and 1.06 and (c) the mean throughput lies between 63543 to 65277.

5.8.3 The Factors Influencing the Response Time

In order to verify whether there is a relationship between response time, hits/sec and throughput, we assume that such a relation, if exists, be a linear one. The response time is assumed as criterion variable. The Hits/sec and throughput are assumed to be predictor variables. The scatter plots of the response time vs. hit/sec and throughput are shown in fig.5.47 and fig.5.48 respectively.

The two scatter plots with their respective regression line shows linear relationship. Greater the value of hits/sec, more will be the response time. In a similar way, the greater the value of throughput more will be the response time.

To examine the combined effect of throughput and hits/sec on response time, we performed multiple linear regression tests. The test is carried out at 95% confidence level. We assumed the null hypothesis (H₀) which is: response time does not depend on hits/sec and throughput. The alternate hypothesis (H₁) is: response time is a function of hits/sec and throughput.

The regression analysis calculates F ratio to be 5.416881, which is significant at 0.05. This provides evidence of existence of linear relationship between response time, hits/sec, and throughput. As such, the null hypothesis may be rejected. This implies that the equation has 95% chance of being true. The analysis also suggests that our model accounts for 28.63% variance on response time. Thus we may infer that the hits/sec and throughput have some influence on response time.

The next chapter of the thesis describes in details the results of the tests performed on PReWebD and PReWebN that have been presented in this chapter and a discussion on the results.
Fig. 5.47: The Hits/sec vs. Response time

Fig. 5.48: The Throughput vs. Response time