EVALUATING HTTP PERFORMANCE FROM STREAMS
USER DOCUMENTATION

- **Team Name:** NAGIOS

- **Team Members:**

  - Atla Prashant
  - Chilukuri, Megh Phani Dutt
  - Garg, Prafull
  - Grandhi, Veera Venkata Santosh S G
  - Kalidindi, Rajeev Varma
  - Kolli, Samuel Sushanth
  - Madala, Sravya
  - Musinada, Suren
  - Naguru, Sriram Prashanth
  - Peddireddy, Divya
  - Rajana, Poojitha

- **Document Type:** User Document

- **Version Number:** Version 1.2

- **Publication Date:** August 24th, 2015

1. INTRODUCTION:

User documentation includes the product documentation i.e. the functionality of each tool, relation between different components in the tool. It gives a clear cut representation of the tool for each input and gives the necessary output.

We focused on the customer who are requiring server process in an optimized way. Monitor, Response time, Bit rate metrics.

The document is partitioned into various sections. Section I deals with System Requirements, Section II deals with Glossary and Abbreviations, Section III deals with Software requirements, Section IV deals with Operation of software.

Release v1.2 on 2015-08-14

PUBLICATION DATE	VERSION	DESCRIPTION	CHANGES
2015-08-14	v1.2	Updated Version	➢ Changes made in front end included.
2015-06-01	v1.1	Updated Version	➢ Step by step procedure of front end is included. ➢ Glossary and abbreviations are included.
2015-05-20	v1.0	Initial Release	

2. GLOSSARY AND ABBREVIATIONS

HTTP: Hypertext Transfer Protocol

It is a protocol at the application level for communication of data between the network elements such as clients and servers.

GUI: Graphical User Interface

An interface which allows the users to communicate with the electronic devices through visual icons. In some cases, it contains audio feedback as well as voice control.

DPMI: Distributed Passive Measurement Infrastructure

This interface is used to read the data stream at various measuring points.

RESTful: Representational State Transfer

An architectural pattern to improve portability, scalability of the system.

RRD: Round Robin Database

This handles the Time series data like data request-response time, Bit rate, standard deviation.

CPAN: Comprehensive Perl Archive Network

This is used for the collection of internet archives.

3. System Requirements:

a) Ubuntu 14.04 version
b) 1 Gb RAM
c) Hard disk 40Gb
d) Processor 32 bit or 64 bit
e) Internet access

4. Software requirements:

Backend:

1. Perl
2. Php5
3. RRD Simple
4. CPAN: Perl Modules
5. Tshark
6. Libcaputils
7. Apache2 server

Perl: Perl is used for developing web applications where it is a high level general purpose language.

Php5: Php5 is a server side scripting language which is designed for web development.

RRD Simple: It is a simple interface to create and store the data in RRD files.

CPAN: CPAN is a collection of internet archives which includes the Perl Modules.

Tshark: It is a network protocol analyser which lets us to capture packet data from a live network. Where it is a terminal oriented version of Wireshark designed for capturing and displaying packets.

Apache2 server: It is an open source license which is a freely available Webserver.

Frontend:

1. Web Browser
2. Lamp Server
3. Text Editor (Gedit)
4. Php5
5. Fetching RRD

Web browser: It is a software application which is used to locate, retrieve and display contents on the web

Lamp Server: It uses as open source web development platform.

Gedit: It is a text editor used in the LINUX operating system.

Php5: Php5 is a server side scripting language which is designed for web development.

Fetching RRD: Fetches the data from the RRD where fetch will analyse the RRD and try to retrieve the data.

MySQL database: It is used to store the values of performance metrics obtained from the log file.

Phpmyadmin: It is a tool written in PHP which is intended to handle the administration of MySQL over the server.

5. Operation of software:

Back End:

a) The user needs to first trace out the HTTP packets using DPMI.
b) The back end code generates the required performance metrics – Request-Response Time, Server Bit Rate, Lost Requests.
c) The metric values that are stored in the database of the provided system architecture.
d) The statistical calculations of the performance metrics can be calculated from the Round Robin Database, which stores and updates the require fields.
e) The tool monitors these three metrics of the server.
f) The user requires that when the request has been sent to the server, the response has to be quick.

g) The user shouldn't have more packet loss.

h) The Request-Response Time and Lost Requests need to have less values. The Server Bit Rate should be high.

Front end:

a) A user needs to register for the first time to view the server statistics like request-response time, the bit rate and lost requests.

b) If he is already a user, he should have a login authentication.

c) The login authentication credentials can be observed in the MySQL database which is connected to the web GUI of the user.

d) The database will store all the require information regarding user and the threshold that should be provided to the servers.

e) If the server threshold value is exceeded than the default value it shows a fault notification to the user. This notification is sent through e-mail.

f) The require graphs can be seen in the Web GUI using RRD.

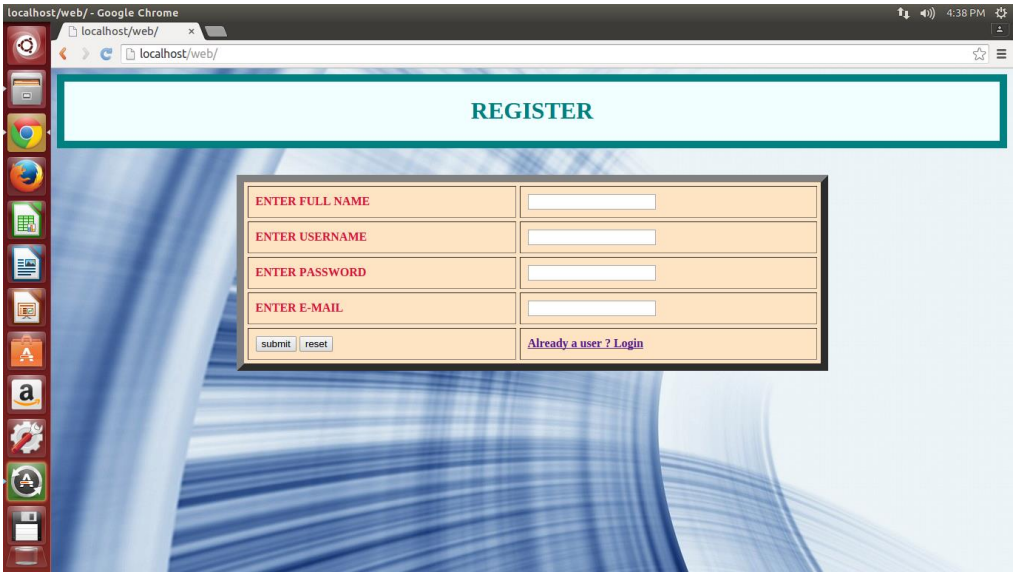
❖ STEP BY STEP PROCEDURE FOR FRONT END:

The below procedure gives step by step procedure for the user to access the frontend of our monitoring tool:

Page 1: Index page

url: http://localhost/web/index.php

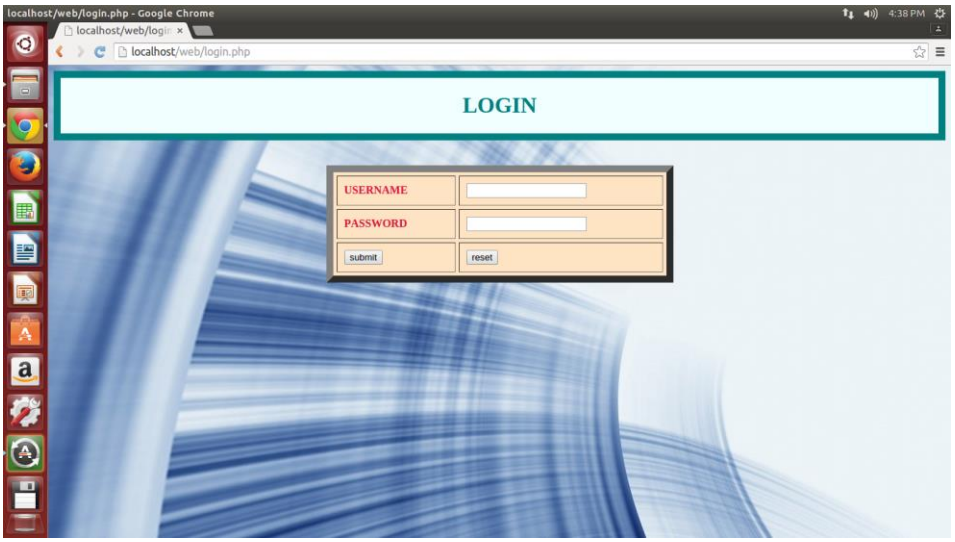
1. Open the web browser.
2. Type the url: http://localhost/web/index.php
3. The user needs to register himself.



Page 2: Login page

url: http://localhost/web/login.php

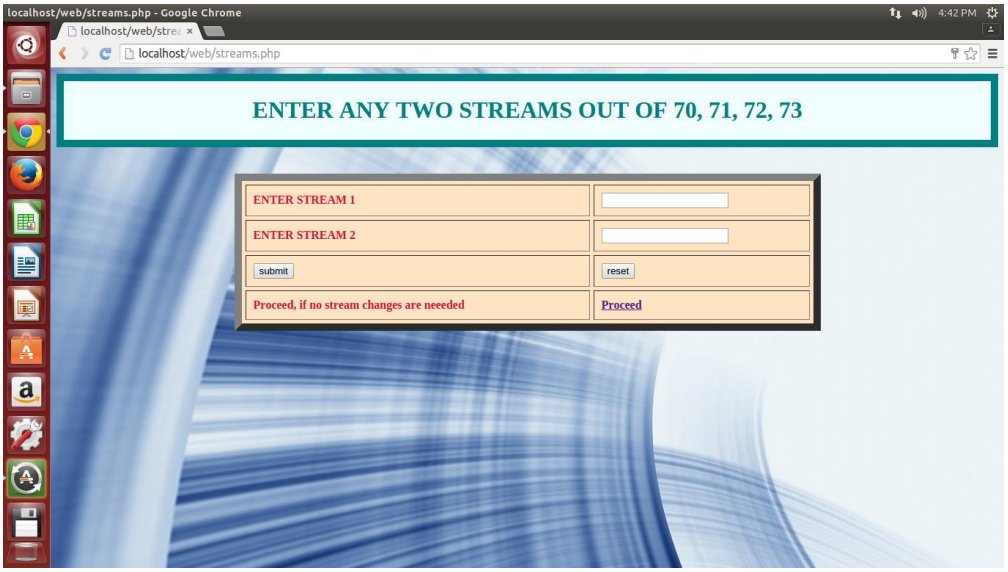
1. The user needs to enter the username and password he entered during registration.



Page 3: Streams page

url: http://localhost/web/streams.php

1. The user needs to enter two streams out of 70, 71, 72, 73 to capture the traffic from DPMI.



Page 4: Home page

url: http://localhost/web/index1.php

1. The homepage consists of three options with a view button below.
    - ✓ Server Metrics
    - ✓ Thresholds
    - ✓ REST API
2. A logout button is displayed on the top right of the homepage.
3. If the user doesn't want to view any of the above options, then they can click the logout button.
4. If the user clicks on the view button below the server metrics they are directed to the Server metrics page (index2.php).
5. If the user clicks on the view button below the thresholds it is directed to the thresholds page.
6. Similarly, the user is directed to the REST API page.

Page 5: Server metrics page

url: http://localhost/web/index2.php

1. In this server metrics page you can observe a box where you can enter the IP address.
2. Beside that IP address box you will see three options.
    - ✓ RRT (Request Response Time)
    - ✓ Rate (Bit rate)
    - ✓ Lost Request

3. Submit button is also present beside the above three options.

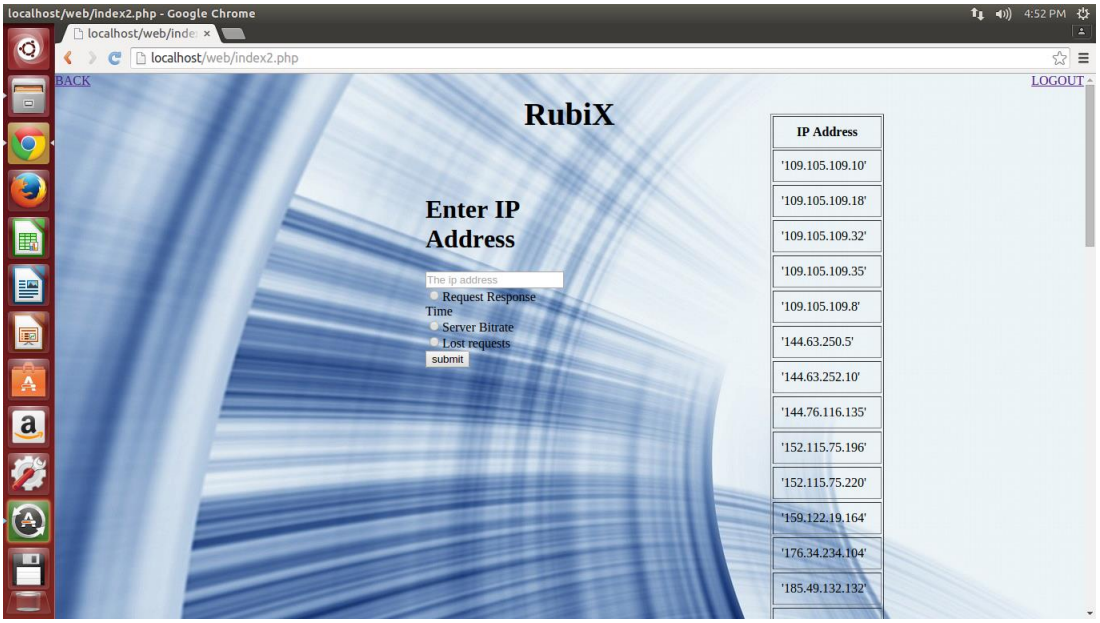
4. The user needs to type the IP address and select the required performance metric and click on the submit button where this page directs to the graph.php. If the user doesn't want to direct for the next page they are provided with two options:

    - ✓ Back button is provided on the top left of the index2.php page.

- **Note**: This home button will direct to the home page index1.php

    - ✓ Logout is provided on the top right of the index2.php.

- **Note:** This logout button will direct to the Login page (login.php)



Page (5a): Graphs Page

url: http://localhost/weeb/graph.php

1. In this page you can observe four different types of graphs with respect to time from RRD.

    - ✓ Hourly Graph
    - ✓ One day Graph
    - ✓ Week Graph
    - ✓ Monthly Graph

2. The graphs are displayed according to the selected performance metric.
3. After observing the graphs the user can direct to any one of the below pages.

    - ✓ Back button is provided on the top left of the index2.php page.

- **Note**: This home button will direct to the home page index1.php

    - ✓ Logout is provided on the top right of the index2.php.

Note: This logout button will direct to the Login page (login.php)

Page 6: Thresholds page

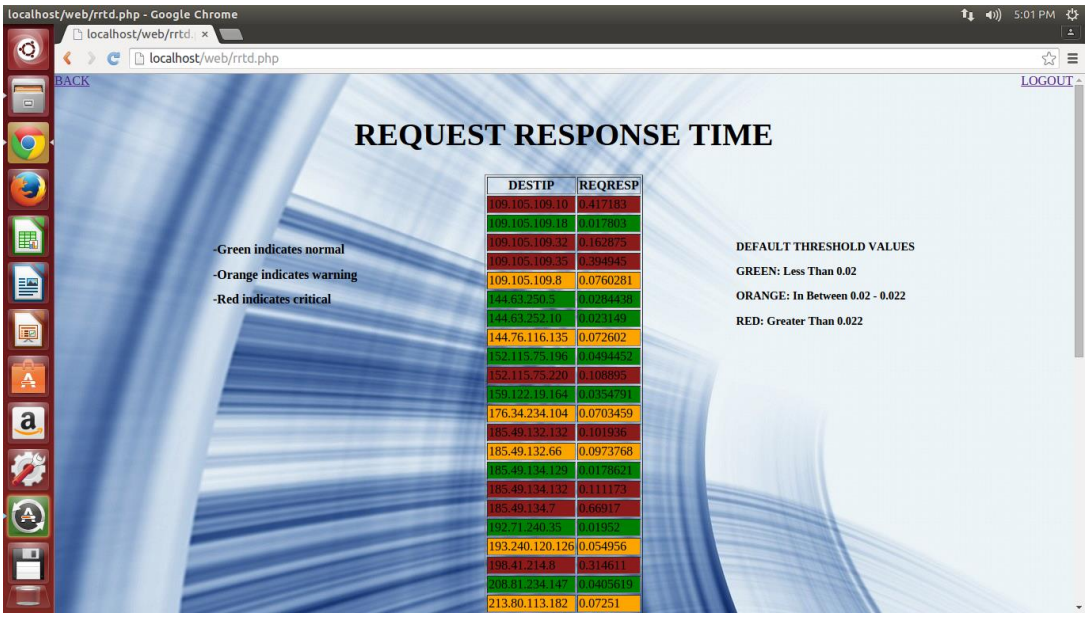
url: http://localhost/web/index3.php

1. For knowing the thresholds, we have to click the view button below the thresholds in index1.php.
2. Then that page is directed to the thresholds page where you will find two options to select the limits for the thresholds:
   - ✓ Default
   - ✓ User-defined

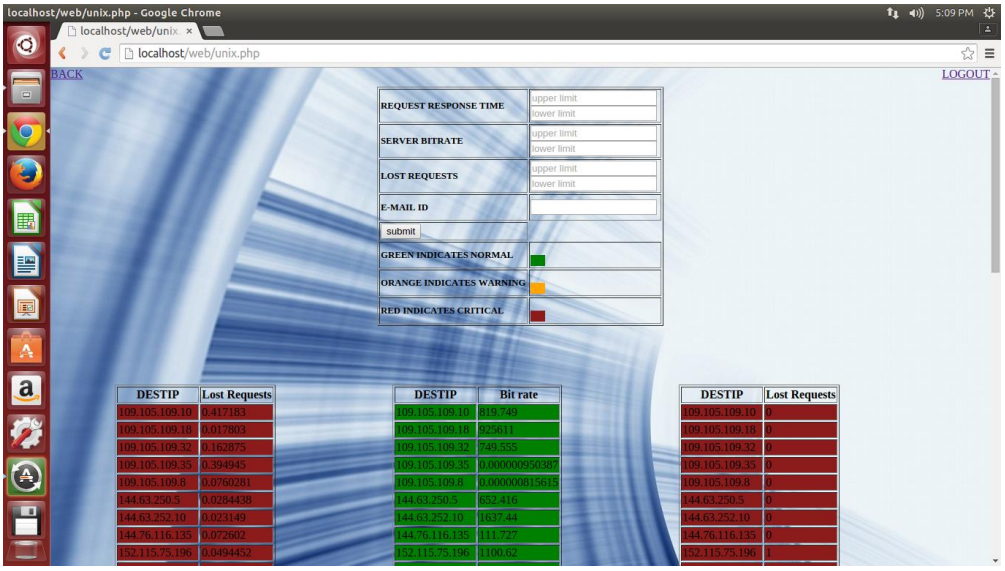
Page 6(a): Default page

url: http://localhost/web/rrtd.php (for Request-Response Time)
url: http://localhost/web/sbrd.php (for Server Bit Rate)
url: http://localhost/web/lrd.php (for Lost Requests)

1. The user can select any one of the above URL to view the desired performance metric.
2. The threshold levels are indicated suing three different colors: Green, Orange, Red for the respective values.
3. An email notification is sent about the status where the email is already stored in the database.

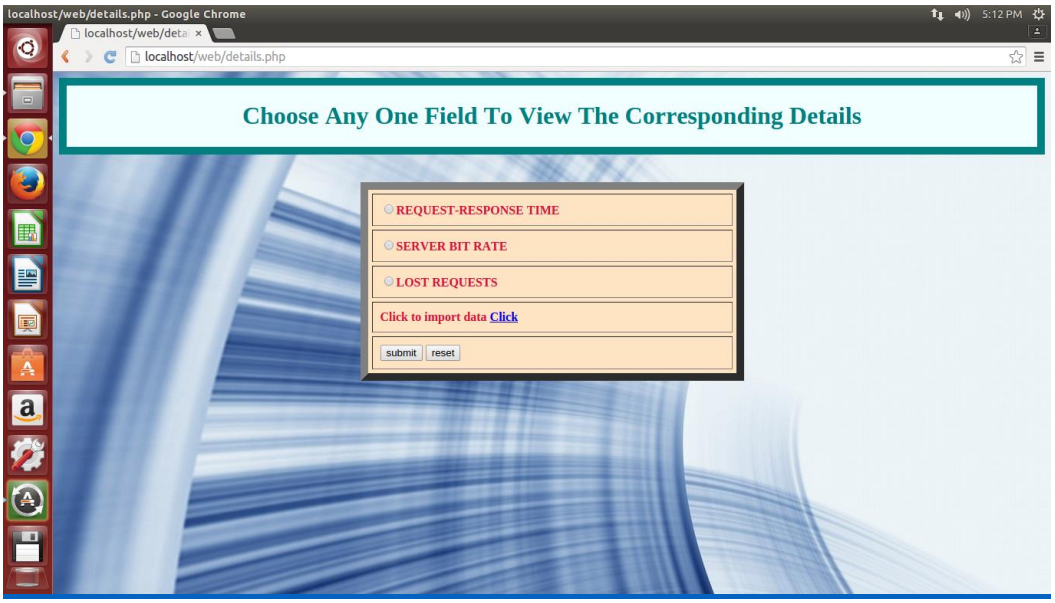


Page 6(b): User-defined page
url: http://localhost/web/unix.php



1. If users click on the user-defined link, they are redirected to a page where they will observe a table, where the users can themselves allocate their own limits and it also shows all the three different states.
    - ✓ Normal
    - ✓ Critical
    - ✓ Warning
2. The three different states are representing the status of the particular limits.
3. An email notification is sent about the status where the email is already stored in the database.

Page 7: REST API
url: http://localhost/web/details.php



1. The user can select any of the three performance metrics to view exporting data through a REST API.

   url: http://localhost/web/rest2.php/?value=REQRESP (for Request-Response Time)
   url: http://localhost/web/rest2.php/?value=BITRATE (for Server Bit Rate)
   url: http://localhost/web/rest2.php/?value=LOSTREQ (for Lost Requests)

2. He can select importing of data, to import files through a form page.
   url: http://localhost/web/uploadform.php (for .csv file)
   url: http://localhost/web/jsonform.php (for .json file)

- **Note:**
- The user should save all the files in var/www/html in "web" folder
- sudo chmod -R 777 <filepath> is the command for giving permissions.