Analysis of Image Processing Results

By
Ravi Jain
(Roll No. 2013167)

SUPERVISOR(S):

EXTERNAL:

Mr. Radhish Ayyappan
Project Manager
Canon ISDC, Bangalore

INTERNAL:

Prof. Ayan Seal
Professor
PDPM IIIT DM Jabalpur

Computer Science Engineering (B. Tech 2013)

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, DESIGN AND MANUFACTURING JABALPUR

7th Report

(10th October 2016 – 21th October 2016)

Dashboard Generation -

Introduction –

- The aim of this project is to design a dashboard for certain csv(comma separated value) and log files, which would display details of these in addition to certain graphs for indicating kernel execution times.
- The python program takes as input a csv file, its corresponding log file, median value files (if any) and path to the images which can be viewed in the dashboard.
- The skills used for the development of this dashboard are python, angularjs, javascript, bootstrap, css, html, matplotlib and jquery.

Procedure –

- The dashboard serves the purpose of displaying details such as kernel execution time, total time etc. on executing certain algorithms over several images and drawing a comparison between different test environment types.
- The set of csv files include details of algorithms such as sum of squared differences, mutual induction, gradient image filter etc.
- The aim was to display the logs of execution of these algorithms over certain images.
- Furthermore, graphs were also made using matplotlib, including comparison plots, box plots.
- Modules such as numpy were used.
- The initial task was to understand use of these tools and install libraries on my system.
- On executing the program, it generates a result.html file which displays the required details and graphs.
- Angularjs and jquery are used for sorting the contents within a table, according to column.
- Bootstrap, css are used for overall styling of the html file.
- Python is the language in which whole code is written.
- Matplotlib is used to generate comparison plots and box plots.

The next issue was to make sure that site runs well on all type of browsers, i.e. Chrome, Internet Explorer and Mozilla Firefox.

After finding certain javascript issues such as certain functions involving use of iframes, issues in jquery load method and ActiveX control issues with graph, steps were taken in order to reduce the use of these and make the site run on all browsers.

Pre – release presentation –

As this site had to be released to other users, the software requirements and setup tools were noted.

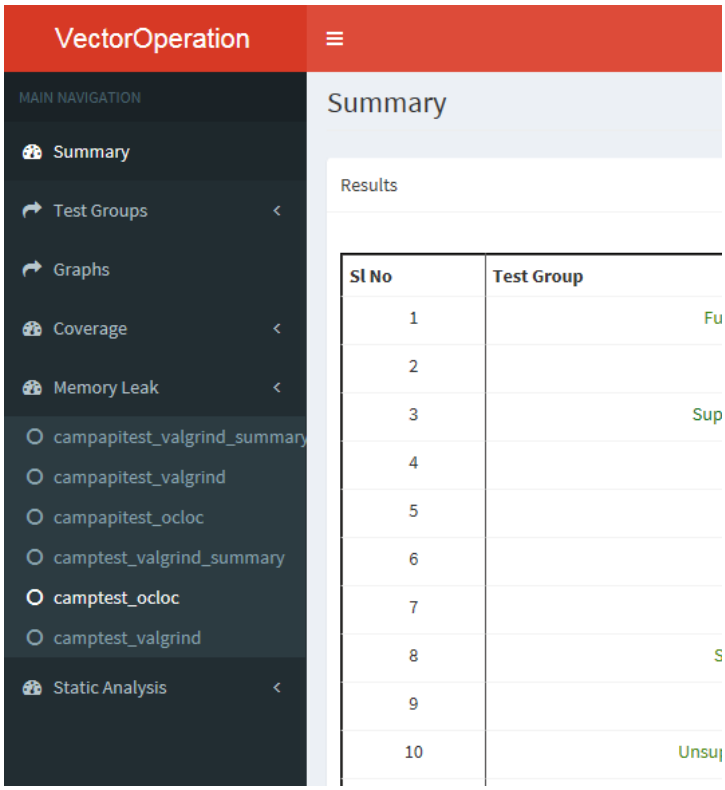
Software requirement -

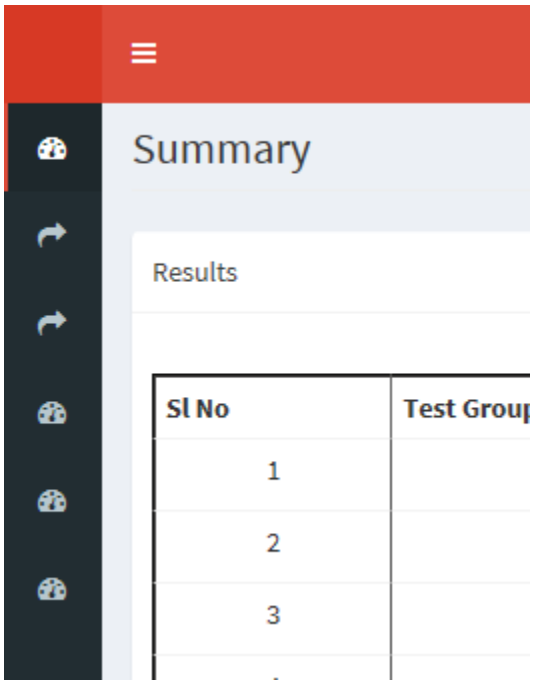
- System must have python >=2.7
- Use command scl enable python27 bash to enable python 2.7
- Packages required are BeautifulSoup4, matplotlib

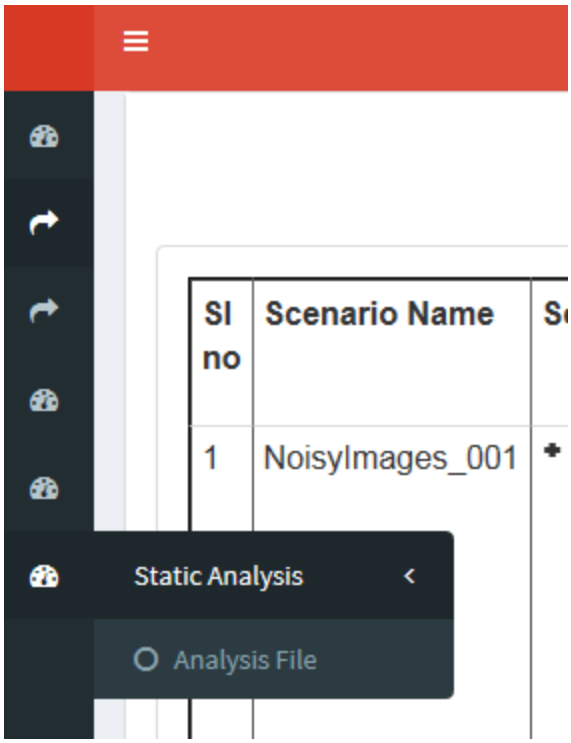
Setup Tools –

- In order to install these packages use pip install package_name or go to https://pypi.python.org/pypi and search for the required packages
- After downloading a package, go to the folder with downloaded content and run python setup.py install

Using certain css tools I learnt how to hide the sidebar and display only icons –

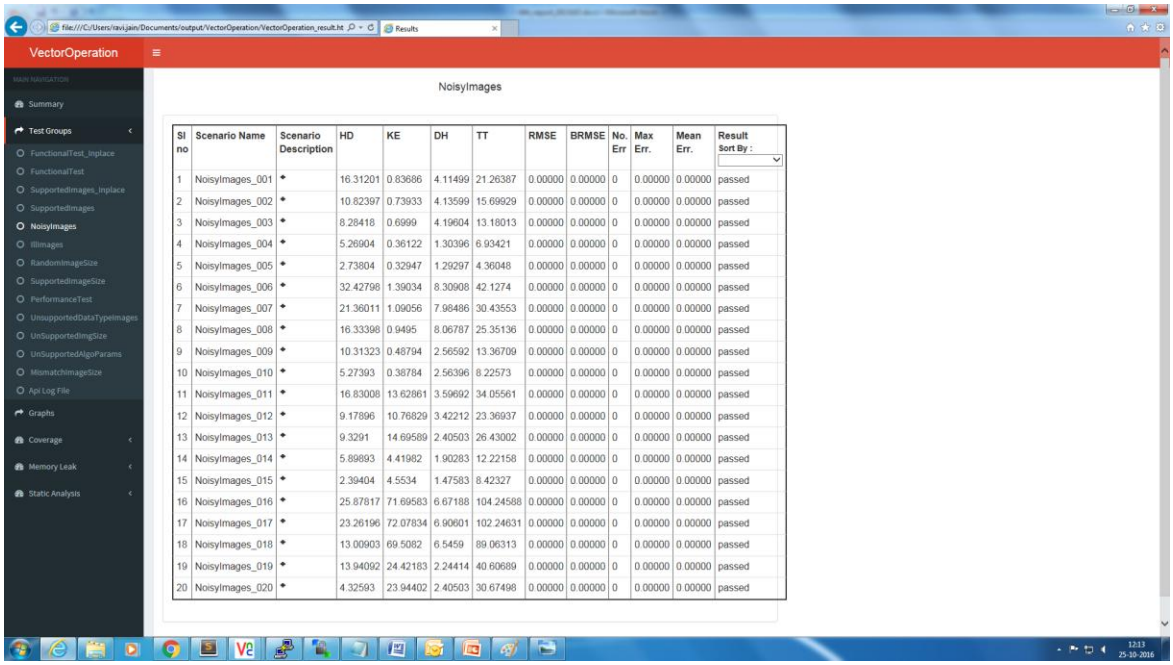

VectorOperation
MAIN NAVIGATION
Summary
Test Groups
Graphs
Coverage
Memory Leak
campapitest_valgrind_summary
campapitest_valgrind
campapitest_ocloc
camptest_valgrind_summary
camptest_ocloc
camptest_valgrind
Static Analysis
Summary
Results
Sl No
Test Group
1
2
3
4
5
6
7
8
9
10


Summary
Results
Sl No
1
2
3

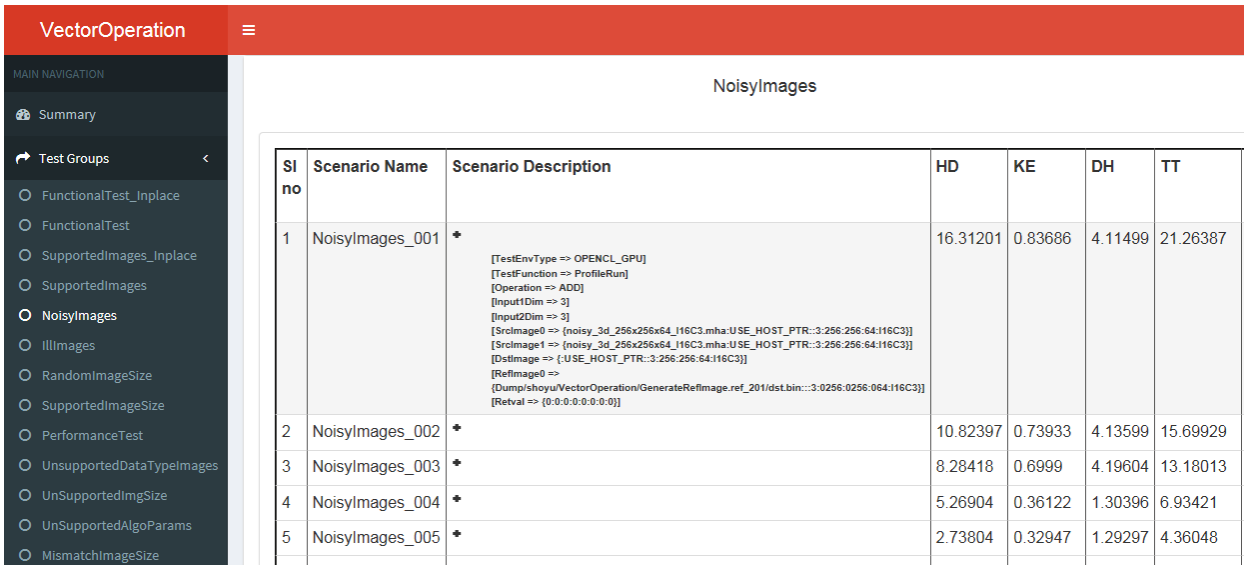


The next step was to check that the code runs for different algorithms, which was done and certain bugs were fixed.

Screenshots of Output generated –



Sl no	Scenario Name	Scenario Description	HD	KE	DH	TT	RMSE	BRMSE	No. Err	Max Err.	Mean Err.	Result Sort By :
1	NoisyImages_001	+	16.31201	0.83686	4.11499	21.26387	0.00000	0.00000	0	0.00000	0.00000	passed
2	NoisyImages_002	+	10.82397	0.73933	4.13599	15.69929	0.00000	0.00000	0	0.00000	0.00000	passed
3	NoisyImages_003	+	8.26418	0.6999	4.19604	13.18013	0.00000	0.00000	0	0.00000	0.00000	passed
4	NoisyImages_004	+	5.26904	0.36122	1.30396	6.93421	0.00000	0.00000	0	0.00000	0.00000	passed
5	NoisyImages_005	+	2.73804	0.32947	1.29297	4.36048	0.00000	0.00000	0	0.00000	0.00000	passed
6	NoisyImages_006	+	32.42798	1.39034	8.30908	42.1274	0.00000	0.00000	0	0.00000	0.00000	passed
7	NoisyImages_007	+	21.36011	1.09056	7.96486	30.43553	0.00000	0.00000	0	0.00000	0.00000	passed
8	NoisyImages_008	+	16.33398	0.9495	8.06787	25.35136	0.00000	0.00000	0	0.00000	0.00000	passed
9	NoisyImages_009	+	10.31323	0.48794	2.56592	13.36709	0.00000	0.00000	0	0.00000	0.00000	passed
10	NoisyImages_010	+	5.27393	0.38784	2.56396	8.22573	0.00000	0.00000	0	0.00000	0.00000	passed
11	NoisyImages_011	+	16.83008	13.62861	3.59692	34.05561	0.00000	0.00000	0	0.00000	0.00000	passed
12	NoisyImages_012	+	9.17896	10.76829	3.42212	23.36937	0.00000	0.00000	0	0.00000	0.00000	passed
13	NoisyImages_013	+	9.3291	14.69589	2.40503	26.43002	0.00000	0.00000	0	0.00000	0.00000	passed
14	NoisyImages_014	+	5.89893	4.41982	1.90283	12.22158	0.00000	0.00000	0	0.00000	0.00000	passed
15	NoisyImages_015	+	2.39404	4.5534	1.47583	8.42327	0.00000	0.00000	0	0.00000	0.00000	passed
16	NoisyImages_016	+	25.87817	71.69583	6.67188	104.24588	0.00000	0.00000	0	0.00000	0.00000	passed
17	NoisyImages_017	+	23.26196	72.07834	6.90601	102.24631	0.00000	0.00000	0	0.00000	0.00000	passed
18	NoisyImages_018	+	13.00903	69.5082	6.5459	89.06313	0.00000	0.00000	0	0.00000	0.00000	passed
19	NoisyImages_019	+	13.94092	24.42183	2.24414	40.60689	0.00000	0.00000	0	0.00000	0.00000	passed
20	NoisyImages_020	+	4.32593	23.94402	2.40503	30.67498	0.00000	0.00000	0	0.00000	0.00000	passed



Sl no	Scenario Name	Scenario Description	HD	KE	DH	TT
1	NoisyImages_001	[TestEnvType => OPENCL_GPU] [TestFunction => ProfileRun] [Operation => ADD] [Input1Dim => 3] [Input2Dim => 3] [SrcImage0 => {noisy_3d_256x256x64_I16C3.mha:USE_HOST_PTR::3:256:256:64:I16C3}] [SrcImage1 => {noisy_3d_256x256x64_I16C3.mha:USE_HOST_PTR::3:256:256:64:I16C3}] [DstImage => {:USE_HOST_PTR::3:256:256:64:I16C3}] [RefImage0 => {Dump/shoyu/VectorOperation/GenerateRefImage.ref_201/dst.bin:::3:0256:0256:064:I16C3}] [Retval => {0:0:0:0:0:0:0:0}]	16.31201	0.83686	4.11499	21.26387
2	NoisyImages_002		10.82397	0.73933	4.13599	15.69929
3	NoisyImages_003		8.28418	0.6999	4.19604	13.18013
4	NoisyImages_004		5.26904	0.36122	1.30396	6.93421
5	NoisyImages_005		2.73804	0.32947	1.29297	4.36048

Due to company's terms and conditions, can't share all images.

Tools Used -

- Python 2.7
- Matplotlib
- Gedit, sublime
- Jquery, javascript
- Angularjs
- bootstrap, css, html
- beautiful soup

Conclusion –

- The readings and output were successfully obtained.
- Certain observations were made regarding these operators.
- The python programming code was able to generate the required files.