Arista – Benchmark Dashboard

Requirement Document

Author: Bijan Mishra (Happiest Minds Technologies)

Bijan.mishra@happiestminds.com

Contents

[Change Log 3](#_Toc453004968)

[Current System: 4](#_Toc453004969)

[High Level Requirements: 5](#_Toc453004970)

[Low level Requirement Split: 5](#_Toc453004971)

[Query Log: 7](#_Toc453004972)

# Change Log

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SR | Description | Date | Change by | Remarks |
| 1 | Draft | 02-06-2016 | Bijan Mishra | Creation of initial draft |
| 2 | Query Log Update | 07-06-2016 | Bijan Mishra | Updated query log to capture Chandra’s comments from his mail and our response to it. |
| 3 | Query Log Update | 15-06-2016 | Radhey | Updated query log to capture Chandra’s comments from his mail and our response to it. |

# Current System:

URL: <http://benchmark>

Script: /src/Artest/www/ArtestCgi.py

Database Name: benchmark

Tables used:

1) Table name: Benchmark

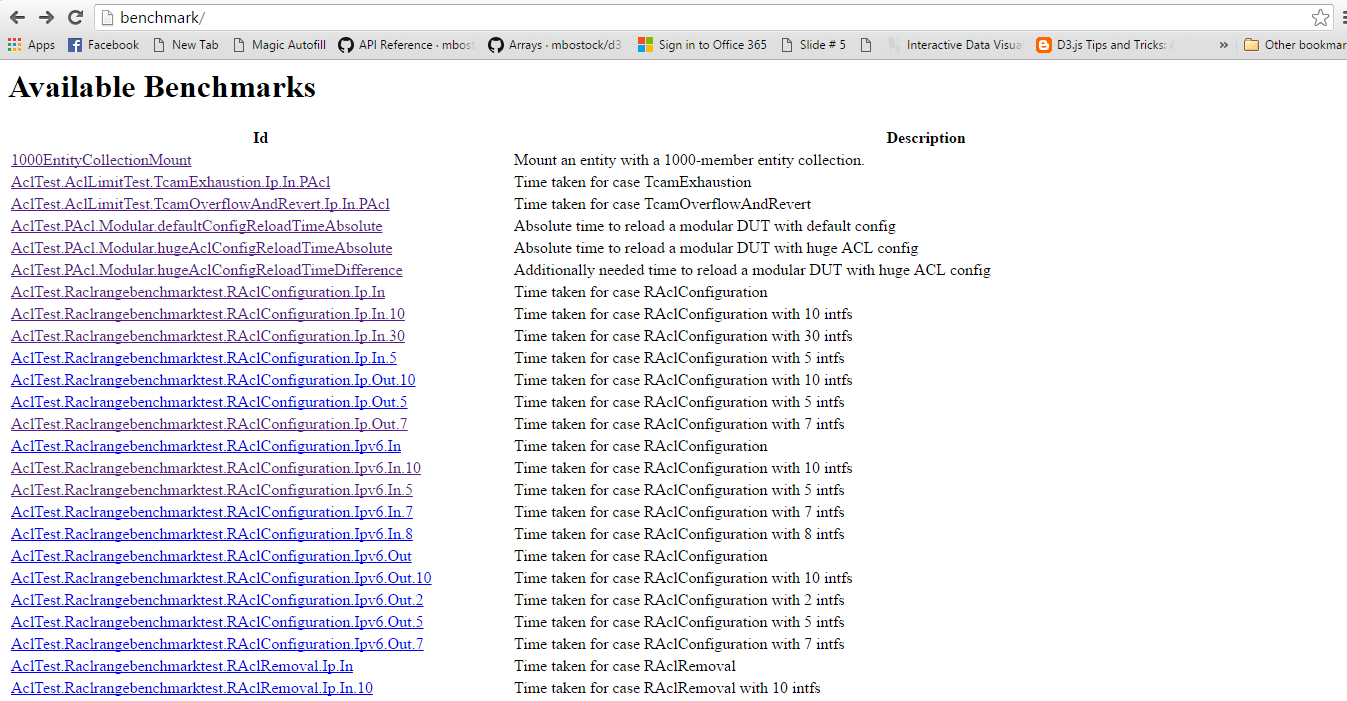
Column names: id (varchar), description (text)

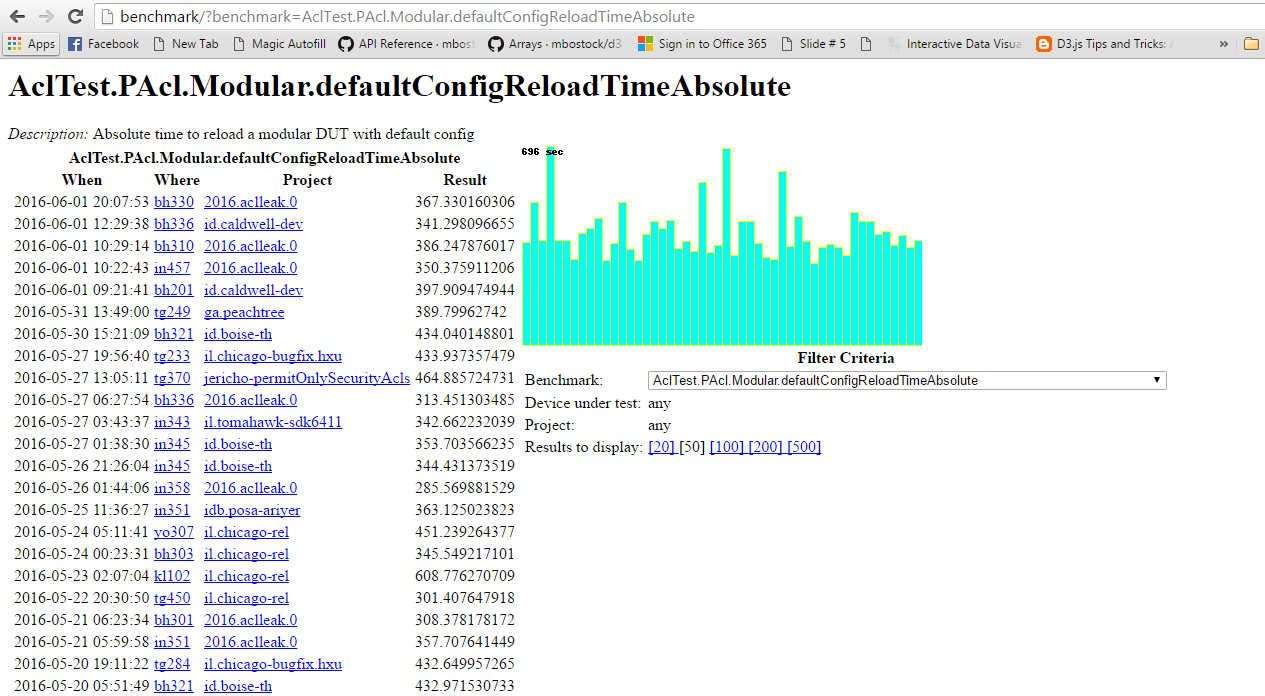
2) Table name: run

Column names: benchmark (varchar), result (double), dut (varchar), project (varchar), release (varchar), client (varchar), changeNum (integer), testTime (timestamp)

Description: Bench marking of the parameters how they behaving.

Pic:





# High Level Requirements:

* Dashboard Visualization of the existing Benchmark Graphs.
* Deliver as a docker container and maintain the code in github.
* Node.js implementation of the current functionality.
* Two level drop down menu to select the id, the first level being the part of the string before the first "." and the second level being the full strings for that.
* Once select a benchmark, the graph and the detailed data should show up on the same page instead of jumping to a new one as the current implementation does.

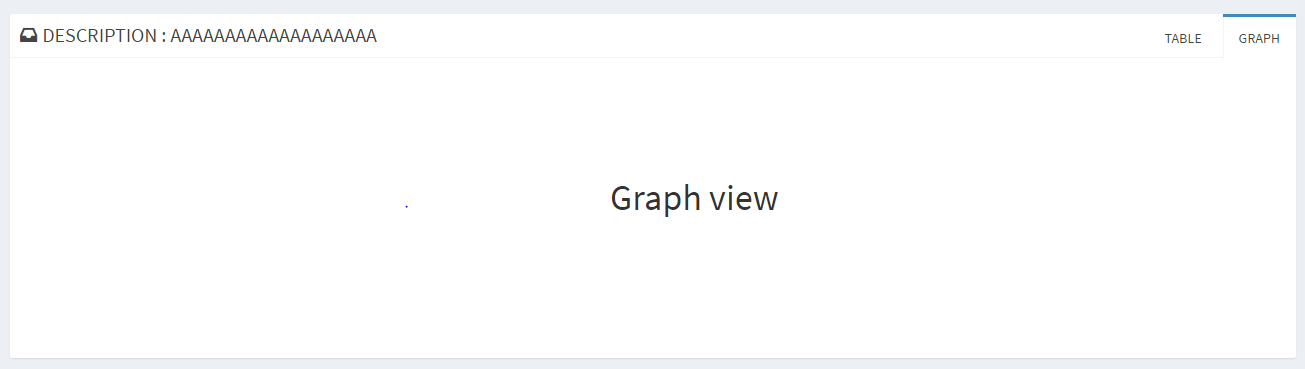
# Low level Requirement Split:

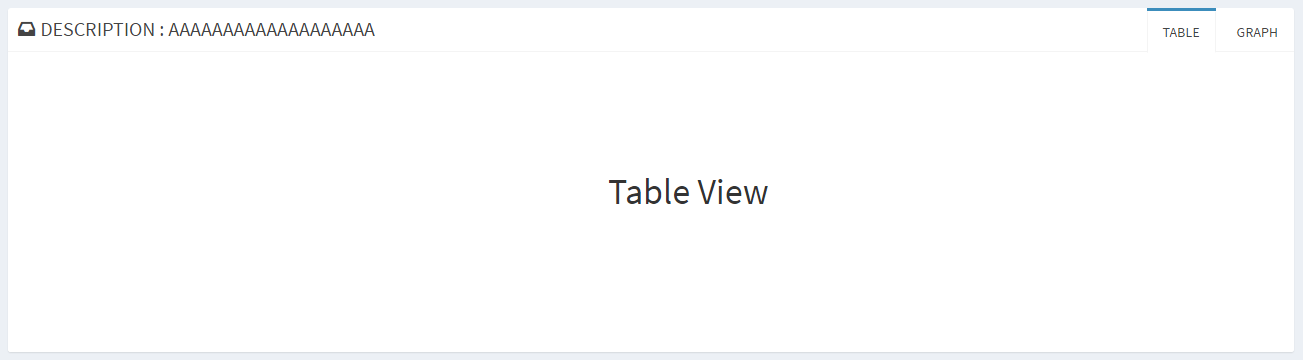
DATA DROP DOWN:

* First drop down will contain the string before the first “.” in the benchmark id, with a search option for easy access.
* Second drop down will contain full benchmark name related to the first drop down. Search option will be there for easy find.
* Initially second drop down will be disabled. Once first one is selected, second one will be enable as per the first drop down.
* Once the selection happens then, a “SHOW” button will enabled to show the graph for the selection.
* If benchmark id does not contain the “.”, then second drop down will be disabled and “SHOW” button will be enabled.
* All the drop down values will be sorted alphabetically

DATA REPRESTATION:

* A tab view will appear below the drop downs, it will contain two tabs one for ”GRAPH VIEW” and one for “TABLE VIEW”. Further details about the two views is given in the following section.
* Data values Changes when “SHOW” button will be clicked.
* Values will be sorted date wise.
* Tab option will be provided for either of the views.
* Description of the benchmark will show at the tab header.
* Demo Representation of tab and description:





GRAPH VIEW:

* Data matrix :- X-axis : WHEN(Date and time),Y-axis : Result
* Graph type : Vertical bar graph
* On mouse hover: WHEN, WHERE, PROJECT, RESULT will show for that bar.
* On load, 50 values will be shown. “50” is an indicative number. Optimal number will be decided during design phase based on the look-and-feel. We will provide options for user to change this number at run time.

TABLE VIEW:

* It will be a tabular view of the data.
* Column values : WHEN, WHERE, PROJECT, RESULT
* Every Column value header will have a sort option for easy data find.
* All the values will be shown in pagination.

Click on any table row will highlight the graph bar for that in their respective view and vice-versa.

Data Filter:

* Table and graph view can be filtered by :
  + When : Date range
  + WHERE : Host name (Multi select)
  + PROJECT : project name (Multi select)
  + RESULT: Range with minimum and maximum run time in seconds.
  + Data values count like (20,50,100,200…) depending upon data values
* Once Filter is applied the graph will change as per the filter selection.
* Refresh button will be there to get the updated values.
* New data will updated to local data base after every 10 minutes.

# Query Log:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SR | Query | Raised By &  Date Raised | Response | Remarks |
| 1 | Do we need an explicit show button? Can't just selecting from the drop down trigger the chart to be displayed | 06-06-16  Chandra | [07-06-2016 : Bijan] - Doing it without ‘show’ button is possible.  As the second dropdown has a very long text, it is possible that user selects in-correct data which will also trigger the graph. Hence he may need one opportunity to verify that he has selected the correct option. Considering this scenario we planned a button so that user can verify the selected value and then click the graph generation. Either we can decide it now, or we decide this during implementation. Kindly let us know if any further queries. |  |
| 2 | What part of the data do you plan to refresh? All the data points might be too much | 06-06-16  Chandra | [07-06-2016 : Bijan] - On first load we will fetch 50 latest records of those benchmarks which are active from last one year from MYSQL database to SQLLite. After that, every 10 mins we will run a job to get the new records and update it in SQLLite. When user clicks on Refresh button, we will display the latest state of records from SQLLite along with last updated timestamp. Kindly let us know if any further queries. |  |
| 3 | What library are you using to draw the graphs? Have you look at the "Google Charts" API? | 06-06-16  Chandra | [07-06-2016 : Bijan] - We are using D3.js D3.js provides more modifications and options to generate elegant graphs which is limited in Google Charts. Kindly let us know if any further queries. |  |
| 4 | Dropdowns should have benchmarks those uploaded data into the database in the past year.  I.e., if a benchmark has not added a new entry in the last year we should ignore it. If it did add data in the last year we should have access to the historical data too. | 08-06-16  Chandra | [15-06-16 : Radhey] – In the Dropdowns, we will be having only names of those benchmarks which are active in last one year.  By default, latest 50 entries of those benchmarks will be cached in SQLite locally. But on user demand, user can have access to entire records of the benchmarks he/she wants to see, (This data will be fetched only when user wants it explicitly.) |  |
| 5 | We should be caching the benchmark names and recent data like the last 50 entries in SQLite and if more data is needed, we should be fetching it from the MYSQL DB in the server. | 08-06-16  Chandra | [[15-06-16 : Radhey] - we will be fetching benchmark names and the latest 50 entries of those benchmarks which are active in last one year and will be storing it in SQLite. And if more data is needed, we will be firing a query based on, for which benchmark data is needed and will be fetching it. |  |
| 6 | What do we mean by past/last year? Either last 365 days (OR) last full year I.e., Jan to Dec 2015 | 10-06-16  Bijan | [10-06-16: Chandra]  Here Last/past year means, one year from current day. I.e., last 365 days |  |