Data Analytics Web Application for Job Runs within Batch-Scheduling Software Control-M

A Technical Report
In CS 4991
Presented to
The Faculty of the
School of Engineering and Applied Science
University of Virginia
In Partial Fulfillment of the Requirements for the Degree
Bachelor of Science in Computer Science

By
Ishan Koroth

December 7, 2023

On my honor as a University student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments.

ADVISORS

Rosanne Vrugtman, Department of Computer Science

Data Analytics Web Application for Job Runs within Batch-Scheduling Software Control-M

CS4991 Capstone Report, 2023

Ishan Koroth
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
ik4vrb@virginia.edu

ABSTRACT

VISA needed job run data from Control-M, a batch-scheduling software to be analyzed for the frequency with which automated processes~~,~~ developed by various technical teams failed, ran for a long time, or started running around midnight. As an intern, my solution was to develop a script that would run every month or less, parse Control-M job run data, execute data analytics, and develop a web application presenting the data to be used by co-workers and senior management. To develop the script, I used Python, the best language for data-parsing through csv files. The data would be split by months and the jobs were sorted by the high level qualifier, or HLQ, of the team that implemented the process. Once the data had been analyzed, a web application would be implemented in node.js where graphs and data sorting options would be implemented to allow the user to view the data in different ways. Ultimately, the web application presented the particular HLQ where a majority of the jobs failed, enabling senior management to get the workload automation team to resolve the problem. Future work includes better script implementation adding features for greater combinations of data representation and implementing a way to dynamically import Control-M data into a database so the data can be continuously updated.

1. INTRODUCTION

VISA has a Global Operations team responsible for monitoring jobs that run on either Control-M or the DPS mainframe system. This team also handles failures by either resolving or reassigning them to different teams that work on the jobs to be fixed. Without this team, the operations and infrastructure of VISA would collapse as a result of overwhelming failures in the system.

As various jobs start migrating to be run on Control-M, issues with job failures arise, some of them requiring a specific prodecure to be handled. Over time, some of the process handling has become automated thanks to the efforts of the Workload Automation team. However, there has not been a way to quantify the job runs, failures, and other data that could be useful for future automated solutions handling jobs that need more attention.

To address this lack of data analytics and quantification, a project had initiated to extract Control-M data for job runs in order to then present the job run data in a web application that allows the user to plot the job run data against failure count and other factors. This was done in collaboration with a Senior Systems Engineer who had helped set up the environment for the Python script runs and the Node.js web application to make this project possible.

2. RELATED WORKS

A framework was needed in order to develop the design for the web application. One idea was to use Node.js [1] to develop a custom web application that can make use of libraries to make graphs and features for data manipulation and representation. Another idea was to use Microsoft Power BI [2] to make use of presets already built-in.

Node.js is an open-source server environment that uses JavaScript, which can be utilized to develop web applications. Thanks to its asynchronous design, Node.js can handle lots of data, which is ideal for this project. For this project, We used Node.js to parse through the resulting data outputted by the Python scripts, and used graphs to compare the data by the number of job runs and failures.

We also considered Microsoft Power BI, as it had been widely used as a tool for developing data analytics pages while providing presets that can save lots of time. Despite the huge efficiency in the workflow, making use of this service would rely on paying for subscriptions, and if support were to be discontinued, then the project would have to be completely redone. The main inspiration from this product was implementing interactivity between the user and the data. This interactivity principle would then be implemented in the Node.js web application, as the user would be able to see the different representations of data.

3. PROJECT DESIGN

To develop the web application, the following steps were executed to accomplish a sequential objective towards the big picture:

3.1. Extract Data

Control-M kept track of job runs from first to last. The data was then extracted into a csv file for parsing.

3.2. Develop Python Script

The data extracted was very large and required a script to parse through it fast and efficiently.

3.3. Add Data Analytics

Once parsing was successful, specific data analytics, such as job runs and failures, were executed and output csv files reflected such data.

3.4. Set Up Web Application

Using Node.js, we developed a user interface on a website where data analytics produced from the script would be displayed.

3.5. Incorporate Data Analytics into Web Application

Once the template was set up, the analyzed data was read from a file directory and displayed on a graph.

3.6. Add More Data Representation/ Features

With the data available for manipulation, more features, like selecting time period, data analysis, or graph type, were added onto the web application to give the user more features for displaying the data in different ways.

4. RESULTS

Once the prototype was finished, it was presented to senior management and co-workers for review. They verified certain jobs that failed frequently and found the data analytics accurate from their experiences dealing with issues in production. Feedback regarding the web design, was generally positive and constructive. Further suggestions were provided, such as graphing the total job failure accumulation over a period of time and being able to change the type of graph for viewing the data.

5. CONCLUSION

By accomplishing the purpose of quantifying data from job runs in Control-M, the web application will eventually be established as a foundation for further data analytical

applications. The Python script will be run each time the Control-M data is updated to parse through data, which will then be displayed in various ways on the Node.js web application. In doing so, my co-workers and senior management can look into the quantitative data and begin to address common issues that seem to be occuring with the open systems batch jobs being monitored in Control-M.

6. FUTURE WORK

The data analytics web application for job runs will be useful in the future as it will eventually be integrated to a dynamic process that will automatically update the Control-M data being presented. Other features, like changing the display graph type and displaying certain intervals of time, will be implemented to add more functionality on the user's end. Last, there will be more data coming from other batch-scheduling software, like the DPS mainframe, so the data analytics web application will come in handy to accommodate data from that end, as well.

7. ACKNOWLEDGMENTS

First, I would like to thank Eric Flanagan, director of Global Operations Core Services and my manager at the time, who got me into the project's initial development phase. I would also like to thank David Rankin, the director of Workload Automation, for enabling the integral relationship between Global Operations and the Workload Automation teams. Of course, the special acknowledgement goes to Sanjeev Rana, the Senior Systems Engineer I was working with to deploy the project. Without his support in managing the project and playing a beneficial role in my hiring, I would not have been able to contribute what I had done for the team, and this web application would still be pending on a to do list.

I would also like to thank my co-workers for their collective support in making my internship experience meaningful with good amounts of insight: Chuck, Cindy, Fawad, Hassan, Joey, CJ, Juliana, and Jack.

REFERENCES

[1] Harikrishna Kundariya. 2023. Why you should use node.js for data science projects? (June 2023). Retrieved September 29, 2023 from https://www.aiplusinfo.com/blog/why-you-should-use-node-js-for-data-science-projects/

[2] Joel Jacob. 2023. An introduction to visual data analytics with Power Bi. (July 2023). Retrieved September 29, 2023 from https://www.projectcontrolacademy.com/visual-data-analytics-power-bi/