An Automated Memory Analyzer Dashboard in AWS

A Technical Report
presented to the faculty of the
School of Engineering and Applied Science
University of Virginia

by

Justin Chen

May 8, 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.

Justin Chen

Technical advisor: Rosanne Vrugtman, Department of Computer Science

An Automated Memory Analyzer Dashboard in AWS

CS4991 Capstone Report, 2022
Justin Chen
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
jyc9fyf@virginia.edu

Abstract

An ongoing issue for many software engineering teams is a lack of quick and easily accessible information about product releases. This often leads to missed deadlines and goals due to uncertainty and slow communication regarding product information. The solution we adopted for this issue was to have a central dashboard containing build information, memory in this case, that can be easily accessed by members of the team and any other team dependent on the project's progress. This was done by first having build memory data collected and sent to the cloud in an automated fashion using the Jenkins Server. A user interface with access to this data, Amazon QuickSight in this case, can now continually collect and plot this information into predefined graphs and visuals. The result is an accurate and automated memory analyzer dashboard that can be easily interpreted by engineers working on this product. In the future, we would like to add more directions to enhance dashboard performance and accessibility.

1. Introduction

The problem that this solution addresses is increasingly essential in fast-paced environments where project teams are constantly put under deadlines. In the quickly-evolving technology industry, teams often do not have the leisure to spend hours fetching information when it could be easily automated. The Automated Memory Analyzer Dashboard is a solution that will save engineers hours of work that could be spent on developing features and making progress toward project goals.

2. Background

Dashboards have served the purpose of giving managers information about a business. For example, a business owner looking to track sales of a product would often employ a dashboard to provide quick information at a glance. This data is usually used in combination with a variety of other statistics to give the owner context about product performance and to aid in decision-making for the next steps. Similarly, the Memory Analyzer Dashboard will provide managers with a high-level overview of memory usage in each build release of a product. This will aid them in deciding the steps that should be taken to change or maintain the amount of memory being used in the next release.

3. Related Works

A group of researchers developed a model from which an effective dashboard can be designed. This model, the GQM model, is defined by the goal, questions, and measures the dashboard will be addressing (Janes, et al, 2016) [1]. Similarly, our dashboard's goal is to analyze memory information for the purpose of meeting memory targets. The question that will be addressed is: "How much memory is being used in each product line?" Ultimately, the

dashboard's measure is the memory usage per product line. Now we can assemble this information to create a dashboard that effectively addresses the main goals of our project.

Few (2007), the founder of the consultancy Perceptual Edge, writes about common mistakes in dashboard creation. One major mistake he mentions is that "the effectiveness of faceted analytical displays can be undermined by gratuitous visual effects, decorative display media, and poor use of colors" [2]. In the creation of the Memory Analyzer Dashboard, it is therefore imperative that its visual design must be as minimalistic as possible while still containing information relevant to our goal, which is to analyze memory information to meet memory targets.

Dashboards like ours are important for non-engineers as well. Amid the recent coronavirus pandemic, dashboards have been important in humanity's efforts to return to everyday life. Virus tracker dashboards such as the interactive web-based COVID tracking dashboard created by a group of Johns Hopkins University researchers (Dong, et al, 2020) [3] have allowed everyday people to track the cases in their area to make an educated decision on how to stay safe.

4. Project Design

The main goal of this project is to analyze memory information to meet memory targets. Because it is a dashboard, memory information must be obtained in an automated fashion using some sort of software implementation. A method of processing this information must also be applied in order to organize the data to be visualized. The dashboard will also need a way to store and visualize any data coming in, also in an automated fashion. This gives teams the ability to adapt to downsizing if memory targets shrink and to quickly identify modules that may be causing a spike in memory usage.

4.1 System Requirements

The requirements for this dashboard include:

- Hosting in an easily accessible location so all team members can view and track their progress.
- Allowing editing so one can debug display or data issues.
- Enabling changing or moving of different displays.
- Adding filter and search functionality so users can look for particular product lines or modules.
- Creating separate pages for different software teams.
- Providing functionality for users to request new modules or organizations

4.2 System Architecture

This project uses several tools. To automate the process, we used Jenkins, an automation server, to gather and upload data. To meet the requirements for display, accessibility and filtering functionality, we implemented AWS QuickSight. Using QuickSight, we can organize and display data received from Jenkins in a variety of visualization formats (e.g., bar graphs, line charts, and data points). Furthermore, different display formats can be organized into separate sheets, allowing multiple teams to utilize the same data but from different views. Last, different product lines can be accessed through a search and filter bar and compared directly in the same chart.

5. Results

The Memory Analyzer Dashboard we built fulfills the system requirements initially set by the team. The dashboard allows users to log in to AWS and select the dashboard they would like to use. Once on the dashboard, there are search and filtering

options that allow users to select product lines or modules for comparison. Each chart also displays memory information over time to track progress toward any target changes.

This dashboard is currently being used by the engineering team and is on track to be implemented across several other engineering teams because of the importance of memory usage information in many current projects.

6. Conclusion

Originally, memory information was manually extracted over the course of several days. This dashboard shortens the process by automatically extracting the information several times a day and is ready when an engineer looks up that information. This project is important because of its value in the software development lifecycle. Time is always a major influencer on the speed and quality of an engineer's work. The Memory Analyzer Dashboard has the potential to decrease the time required to develop products by days and to improve quality because of the inherent consistency of the data being produced.

Working on this project has taught me the importance of communication in any workspace. Throughout the project, it was essential for me to communicate and update my mentor and manager on my progress so they can make adjustments to their plans for the project. When working with software engineers, it was also important to communicate time frames and schedules so that projects can be completed in a timely manner.

During this project, I gained experience working with an engineering team in a fast-paced workplace. I walked away feeling much more confident in my development skills and applying what I have learned to more projects in the future.

7. Future Work

While our dashboard has met its requirements, we would still like to add more directions to enhance(?) performance and accessibility by:

1. Implementing a similar dashboard for other product line information (features, battery, performance) to improve efficiency in the product development process.
2. Expanding this project to project teams outside of the modem group, as memory information is also required and applicable to many different non-cellular products.
3. Automating the process for similar dashboards so that graphs and visualizations do not need to be manually populated.

These improvements do not need to be implemented right away and are only plans for further development if this project gains traction in project teams outside of our group.

References

[1] Janes, A., Sillitti, A. and Succi, G. 2016. Effective Dashboard Design. *Research Gate*. Retrieved from https://www.researchgate.net/profile/Alberto-Sillitti/publication/286996830_Effective_dashboard_design/links/57c699e208aec24de0414df1/Effective-dashboard-design.pdf

[2] Few, S. 2007. *Dashboard Confusion Revisited.* Retrieved from http://mail.perceptualedge.com/articles/visual_business_intelligence/dboard_confusion_revisited.pdf

[3] Dong, E., Du H. and Gardner, L. 2020. An interactive web-based dashboard to track COVID-19 in real time. *The Lancet Infectious Diseases* 20, 5 (February 2020). DOI:https://doi.org/10.1016/s1473-3099(20)30120-1