Database Drift Detector

CS 4991 Capstone, 2021

Dev Kumar
Computer Science
University of Virginia
School of Engineering and Applied Science
Charlottesville VA USA
dk9nr@virginia.edu

Abstract

The Database Engineering Services team (DBEST) for Salesforce, an American customer relationship management service cloud-based software company headquartered in San Francisco, designed a program to discern the differences between special SQL Queries called stored procedures (sprocs). These sprocs are saved SQL Queries to be rerun multiple times. Machines have different versions of each sproc, i.e. it is non-standardized. The program ensured that the sprocs were identical to each other or confirmed the differences. There are multiple teams in the Marketing Cloud organization that use and share sprocs and thus this update would ensure that code is standardized between teams.

The project saved time and effort. Before, team members were unsure if their team's sprocs were updated. Now they can access a Splunk dashboard in order to check which sprocs have multiple definitions. It is significantly less susceptible to human error as the process is not only automated but also visualized in a table. The new process will be used for multiple databases across multiple teams. The product is complete in the sense that it does what it aimed to. However, there are plans to not only find the differences between the sprocs but also correct them automatically as well.

1 Introduction

Bill Gates has claimed that "The first rule of any technology used in a business is that automation applied to an efficient operation will magnify the efficiency". The databases in the DBEST team were used to store customer information. Clients included Target and Gap Inc. Multiple teams and external actors interact with these databases. Some run the same SQL queries recurrently and so these queries are stored to save time. A majority of the sprocs are used on multiple databases. However, there was no simple solution to ensuring that the sprocs were identical to each other.

There are multiple benefits to using stored procedures. They provide an

important layer of security between user interfaces and the databases. They also preserve data integrity as the data is entered in a consistent manner. They also improve productivity because the code saved as a sproc only must be written once.

2 Related Work

The most similar product may perhaps be a simple software that finds the differences between two pieces of text documents. Another tool more relevant to the field of Computer Science is git, which is capable of tracking multiple versions of code. However, neither are specific to the database engine the organization used, MySQL and neither were used as guidance for the program.

3 Project Implementation

The project started with using a SQL Agent Job which calls a powershell script every day. Jobs are used to call scripts procedurally according to a set schedule. The scripts run once every night, gathering a list of instances in use by querying a database that stores active instances with “IsActive” set to “True.”

Each instance is looped through and its sprocs are MD5 hashed. The code query itself is the only part hashed. The hashes are then extracted to a remote server. With the help of another team specializing in Splunk, the hashes are then forwarded to Splunk, where Splunk Processing Language is used to form a query to count the number of unique MD5s for each of the sprocs. The Splunk Fundamentals 1 [1] course was used to get familiar with using the software. If a sproc has more than one unique hash, that means there are multiple definitions for it. Then the team manually checks the different instances and figures out ~~what are~~ the differences between the sprocs.

There are four total important pieces of data in the Splunk dashboard, the final product. The first is a report of procedures with multiple unique MD5 definitions. This is the main project feature. Another aspect is the ability to search the name of a sproc and calculate the number of servers that have each of its unique MD5s. If there are many servers with one MD5 and only one with another definition, it is highly recommended to check the latter for its inconsistencies, although ideally all servers should be checked to determine why they have different definitions for their procedures. The third aspect lets the user search an MD5 and output which servers have the MD5. Team members usually know the names of different types of servers and can determine if those servers should have that particular MD5. The final feature is a line graph that illustrates the number of total unique sprocs per a user defined time interval. Optimistically, there is a downward trend, indicating that team members are finding inconsistencies and fixing them. Figure 1 displays the

code used to store sproc information into a PowerShell script. Figure 2 displays how a user may use the final product in order to track different versions of a sproc.

```
$instance=$args[0]
$query = "SELECT @@SERVERNAME AS ServerName,
SPECIFIC_CATALOG AS DatabaseName,
SPECIFIC_NAME AS ProcedureName,
SPECIFIC_SCHEMA AS SchemaName,
CONVERT(Varchar(32),HASHBYTES('MD5',Routine_Definition),2) AS MD5,
'ERROR' AS AlertLevel,
Convert(date, getdate()) as [TimeStamp]
FROM [Utility].[Information_Schema].[Routines]
WHERE ROUTINE_TYPE = 'PROCEDURE';"
```

Figure 1: **SQL Query used to extract an MD5 hash from each stored procedure**

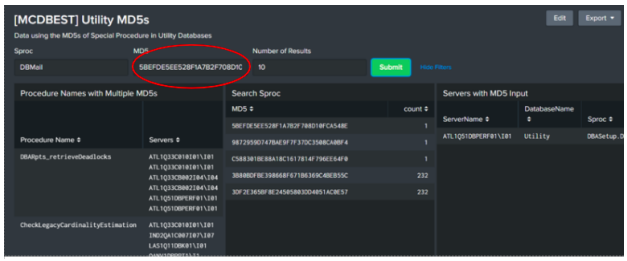

Figure 2: **Final product as demonstrated via Splunk**

4 Results

The product was completed shortly before the end of the internship. The user interface included additional information such as displaying the number of unique MD5s over a user-set time interval. Team managers stated that their team would use the product. A team member asserted that he has been waiting for the process of finding differences between sprocs to be automated for years.

5 Conclusions and Final Thoughts

Even though I returned to the same team, I worked with a completely different set of technology. Last year I primarily worked with C#. This summer I worked with SQL, PowerShell, and Splunk. I could manage SQL and PowerShell reasonably well but I took the Splunk Fundamentals 1 course in order to familiarize myself with Splunk. No one in my team of 20 had much experience with Splunk as well. I also worked on tickets related to the project I worked on the previous summer.

I also had to work with an external team to get Splunk functioning as intended. This was something I did not have much previous experience with. I experienced frustration when progress slowed down drastically while waiting for others. Slack messages were often ignored and updates through the official Kanban dashboard were slow and sometimes done incorrectly. There were many bugs that took multiple days to solve. There were design decisions and multiple languages considered and discussed with team members. For example, I had an implementation using exclusively SQL but was discarded so that Powershell would call SQL and then forward the data to Splunk.

Even though I had a mentor, I also had multiple one-on-ones with different members of the team, either for help or seeing what they were working on. I also had meetings with external members to solve problems such as getting proper authentication or learning about particular features and practices exclusive to Salesforce. Overall, the project was challenging yet doable and I

now have a better picture of the work I may be doing after graduation.

6 Future Work

The main work to be done in the future includes automatically correcting the difference between sprocs. As of now, the user still has to manually check the sproc itself through the MySQL engine after using the program. In a future implementation, the code for each of the sprocs could be displayed in Splunk itself and the user could pick which definition of the sproc is correct. Then the program would automatically update the outdated sprocs.

7 UVA Course Evaluation

The Bachelor of Science in Computer Science aided in being able to complete the project in a timely manner. The material learned in the courses CS2150: Program Analysis and Data Representation and CS4750: Database Systems especially helped with project realization. The former helped with software engineering in general while the latter helped with understanding how to formulate SQL queries and how databases are implemented.

However, two of the greatest factors in finishing the project were the completion of previous internships at a startup, Amazon, and Salesforce. Just like the latest experience with Salesforce, two of the internships before it were both remote. The start-up internship was in-person but utilized ColdFusion which helps connect HTML to PostgresQL which gave the first exposure to database usage in industry. The internships gave valuable insight in how teams collaborate ~~together~~ to build a project used between teams. While the manager remained the same as last summer, another team member acted as the mentor, answering questions and problem solving when needed.

8 Acknowledgments

There were many people who supported the completion of the project. The recruiter, Stacy Schall started the whole experience but team members and other interns contributed the past summer. Particularly, Matt Jones and Patrick Tidrow were the primary points of reference ~~when needed~~ but all team members offered aid when needed.

References

[1] SPLUNK. *Splunk Fundamentals 1.* Retrieved November 28, 2021 from https://www.splunk.com/en_us/training/free-courses/splunk-fundamentals-1.html