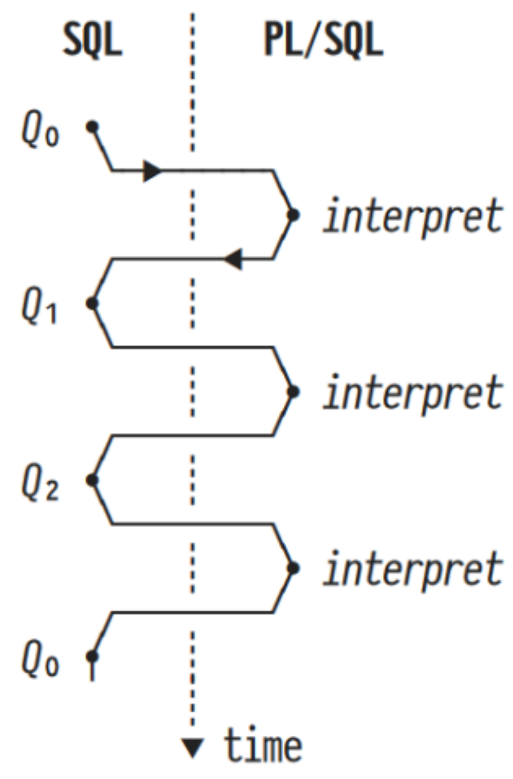
Sniffing PL/SQL Code Smells

Abstract

PL/SQL allows the use of a procedural language in combination with SQL. This lets users implement arbitrary complex data structures such as loops and conditionals. As a result, PL/SQL provides more functionality to SQL. Yet, due to the additional functionality, PL/SQL is also flawed in terms of computational efficiency. This is due to the dual nature of the language, it must be able to switch between the procedural interpreter and SQL as a query language. Code sniffer for PL/SQL will provide a way to find out code inefficiencies by analyzing the code, this can be implemented as IDE Extension as well. Sniffer will ensure predefined standards for code styles and will alert users with performances matrices to help coders see how their PL/SQL code is performing. As PL/SQL is already quite complicated, and there are so many variable combinations of PL and SQL, so its not that easy to standardize the code for future improvements.

Problem

PL/SQL inefficiencies arise due to its dual nature, the transfer of execution of code between procedural language and SQL consumes extra time. For example, an arbitrary user-defined function is designated and invokes PL/SQL. The user-defined function is called to interpret PL/SQL code. Once the function reaches an embedded SQL construct, a switch is done to the SQL database side. As soon as the database side is finished, another switch takes place and the interpretation of the PL/SQL code is resumed. This continues until the operation is completed. The constant back and forth between PL/SQL and SQL leads to inefficient runtime.

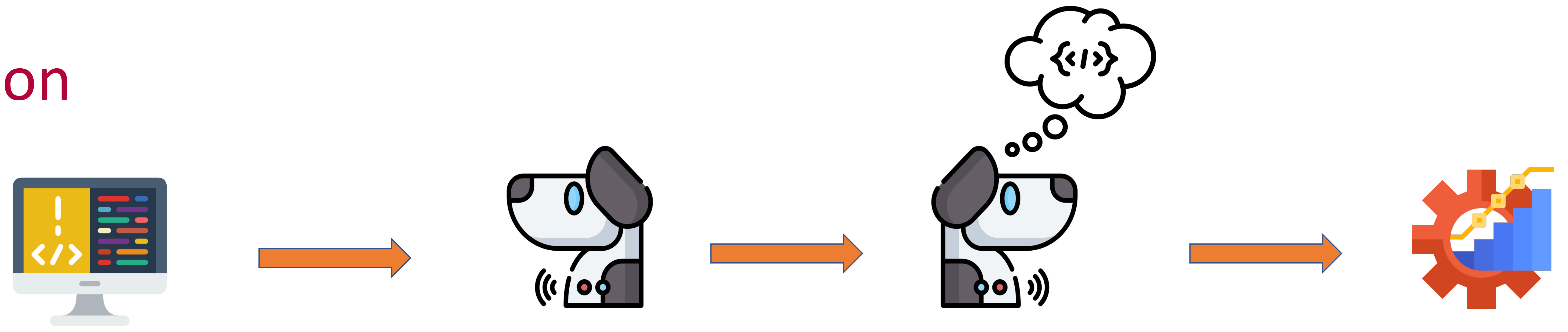


Goal and Research Questions

Main goal of this research will be to find a way to standardize the PL/SQL code, at least for certain types of algorithms or problems and then build a code sniffer that can read the code, analyze the code and then run it to get final matrices and help pointing out the areas where using certain styles or certain queries can help achieve better performance. Following research questions can be derived from this goal:

1. Is it possible to standardize the PL/SQL code at least for limited scope algorithms?
2. How far can we go in standardizing the PL/SQL Code?
3. Will it worth the effort to design standards and code sniffer since PL/SQL is not very widely used anyway?

Solution



Not Optimized Code

Code Sniffer in action

Code Sniffer suggestions

Optimized Code

As much as the problem of inefficient PL/SQL goes, steps can be taken to standardize more and more ways to write the code. For example different data types might have different behavior in PL and SQL and programmer can optimize the process by using the most efficient approach and then that approach can be standardized so PL/SQL code sniffer can compare existing code with most efficient code and suggest if its better to use PL or SQL for certain data types or for certain size of data being transferred. Modern IDEs support extensions, and PL/SQL Code Sniffer can be offered as IDE extension to automate the process of development as well. Code sniffers already exist for other languages such as Jet Brains Code Sniffer for PHP[1] and they do provide performance advantages to the users by applying standardized way of coding where possible. PL/SQL Code sniffer can also help in one of the most inefficient parts of the setup, the switching between PL and SQL.

Muhammad Kamran Ali

Masters Student, Digital Health Center
Hasso Plattner Institute, Potsdam, Germany

E-Mail: Kamran.ali@student.hpi.de

References

[1] https://www.jetbrains.com/help/phpstorm/using-php-code-sniffer.html