Technology Review

Oregon State University
CS 461
2016-2017

Prepared By:
Nic Desilets, James Stallkamp,
and Nathaniel Whitlock

November 11, 2016

Abstract

TODO

CONTENTS

1 Toolkits

1.1 Purpose of a Toolkit

The main purpose of a database toolkit is to assist a Database Administrator (DBA) with analyzing the performance of queries made to a database. The different types of toolkits that can be used with Oracle Database 12c vary widely from PL/SQL queries that can be run directly in the database to complex, fully featured web applications. Likewise, the use cases for these toolkits can range from quick and easy query statement diagnostics all the way to automating the analysis and optimization of parameters within the database to a particular SQL query or multiple queries.

When analyzing queries, being able to effectively quantify and visualize multiple aspects of a query statement is important to a DBA. Some of the information gathered might include the time it took for a query to run, how much memory was allocated to a process handling a query, and the amount of disk input and output operations. This information can then be combined and analyzed in order to identify problematic areas within the database with a particular query or multiple queries.

For example, if you have a particular slow running query, you might use a toolkit to analyze what is happening when the query is being run. After analyzing the output of the toolkit you then find out that the way the database is allocating memory is suboptimal for the query thus resulting in a lot of swap space usage. With this knowledge you can then adjust parameters within the database to allocate more memory to the processes that handle queries.

1.2 Existing Toolkits

1.2.1 Oracle Enterprise Manager

Oracle Enterprise Manager (EM) is already included in the Enterprise edition of Oracle Database 12c. It is intended to be a one stop, all in one application that can be used for analyzing and optimizing query performance. Unlike other toolkits which are comprised of PL/SQL queries that can be executed in the database, EM is a full fledged web application that is packed with many features beyond analyzing query performance.

Some features that are included in EM are: the ability to manage and maintain several databases from one web application; automating repetitive tasks involved with maintaining databases; testing changes before rolling out to production; diagnosing performance problems and automated database tuning. Additional examples of what EM can do are it can determine if indexes will be helpful for performance of a particular table and analyzing sql query statement structure for potential performance issues.

One particularly useful feature of EM for analyzing query performance is a subcomponent called Active Session History (ASH). The ASH tool runs in the background and samples each active session within the database once per second. Each sample for each session contains detailed info about what resources are being used by the session and how the session is using it. This is especially useful for identifying bottlenecks and other performance issues of running SQL queries.

1.2.2 SQLd360

SQLd360 is a tool that is written by Mauro Pagano, an ex-Oracle Database Engineer of about ten years, which analyzes SQL queries. Unlike EM, it will only analyze a single specified query and will not perform any database optimization for you. Instead, it is only intended for performance analysis of a query. SQLd360 is very similar to EMs ASH tool as it operates under the same underlying principle of periodically sampling the session that the query is running in to gather performance metrics. This information can be specified to be output in either html, text, csv, or graphical charts. One important difference is the SQLd360 is provided as a completely free tool while EMs ASH requires additional licensing from Oracle. This makes it particularly useful for users who do not have the required licensing but still need to obtain key metrics from session data. After the tool is finished running, it will dump all of the collected information into a zip file for later analysis.

1.2.3 Snapper

Snapper is another toolkit written by Tanel Poder, an Oracle Certified Master DBA, which also analyzes SQL queries. Similar to SQLd360, it is also provided completely free of use and does not require additional licensing from Oracle. It also operates similarly to SQLd360 in that it uses the databases session information to extract metrics information from running queries. It will also not provide any recommendations about how to optimize queries or database parameters. In addition to sampling data from query sessions, it also gathers data from the V$ and X$ tables within the database which both house extra information regarding query performance. Snapper combines this information together in order to provide a data rich visual representation of how exactly a query is running in the database.

1.3 Creating a Toolkit

Of the three toolkits mentioned above, it might sound like Oracles EM is the obvious way to go since it is very feature rich and is even capable of automatically optimizing SQL queries and database parameters for you. However, in our case we are still going to develop a toolkit of our own that will provide similar functionality. While avoiding reinventing the wheel is typically important when it comes to designing and creating software to prevent wasting time, in our case it makes sense for us to design a toolkit of our own that provides similar functionality to the three toolkits described above. The reason for this is, is that it will force us to become familiar with the inner workings of the database. While we could just use of the already existing toolkits to perform query performance analysis for us, we would probably not be as effective in modifying important database parameters when it comes to memory management, database table partition design, and parallelism. The expected outcome of developing our own toolkit is that not only will we have a custom made toolkit tailored to our needs but we should come out much more knowledgeable about Oracle 12c. This in turn should allow us to become more proficient in customizing the parameters of the database to increase query performance.