Technology Review

HP: Big Data Analytics
Group 18

Alex Schultz
Oregon State University
Corvallis, Oregon

November 22, 2017

Abstract

This document serves to review one article and two technologies essential to the completion of the project prescribed by PageWide Web Press, a division of Hewlett-Packard. *Compression in Oracle - Part 1: Basic Table Compression* - an article that laid the comprehensive foundation on which the project is structured, Oracle Compression - methods for maximizing storage, and Performance Analysis - the methodology of evaluating effects of compression on query performance, are imperative to fulfill client aspirations. At least one alternative to each proposed technology is evaluated and compared, excluding the article. After researching and comparing the technologies to at least one alternative, the team decided to use both Basic Table Compression and Advanced Row Compression and a custom performance analysis toolkit. Alex Schultz, Dylan Davis, and Trevor Hammock conduct research, implement, and analyze all tasks assigned by the project coordinators.

Contents

1 Introduction

The PageWide Web Press printing division of HP receives 350GB of data per day, and they would like to learn how to effectively use compression techniques to reduce the storage footprint while maintaining or improving query performance. It is critical that the team uses appropriate technologies to accurately analyze compression methods and measure performance. Without proper tools, unforeseen problems may arise and create roadblocks, hindering the integrity of the project. The following two technologies are reviewed and compared against at least one alternative: Oracle Compression, and Performance Analysis. The article "Compression in Oracle - Part 1: Basic Table Compression" is evaluated with respect to its contribution and value to the team. The purpose of this document is to evaluate these technologies and discern whether they're optimal for compressing, storing, querying and analyzing data.

2 The Essential Article

2.1 Compression in Oracle - Part 1: Basic Table Compression

Jonathan Lewis is well-known in the Oracle environment, as he has over 20 years experience as a freelance consultant with the Oracle relational database management system [1]. He specializes in physical database design and solving performance issues [1], so he is an incredibly reliable resource. He published this article on January 15th, 2013.

Early in the project, the team was tasked with obtaining a general understanding of data blocks. Upon acquiring foundational knowledge about data blocks, the goal switched to reverse engineering the way in which compression works at the block level. An important thing to note is that the assignment of which we were tasked delves into relatively undocumented territory. It took a couple weeks just to find this article, but it laid the foundation of our understanding.

Jonathan Lewis beautifully describes the mechanics of the way in which Basic Table Compression works at the block level [2]. It was written so well that the current team, essentially foreign to low-level database mechanics, was able to quickly and easily understand what Basic Table Compression does and how it affects blocks. Additionally, he provided insight as to how to read trace files, which are created when a data block is dumped in Oracle Database. This knowledge is absolutely fundamental in the process of understanding and reverse engineering data blocks and compression techniques. He used charts and tables and examples, allowing his ideas to flow and transition smoothly. There is truly no other article currently available that provides superior insight, and that is why it is essential to this project.

3 Oracle Compression Options

3.1 Introduction

The amount of data that enterprises are storing and managing is growing rapidly - various industry estimates indicate that data volume is doubling every 2-3 years [5]. With over 350GB of data introduced to PageWide Web Press' servers each day, it is essential to manage the data in an effective manor. To provide perspective, and assuming there are 30 days in a month, HP's printing division receives approximately 126TB of data each year. The enormous growth in the volume of data makes storage one of the biggest cost elements of most IT budgets [5]. To avoid upgrading server hardware to cope with increasing storage demand, the team will analyze compression methods to maximize efficiency. This section examines Basic Table Compression and Advanced Row Compression. There is a section regarding Advanced Network Compression, but it is simply a *potential stretch goal*, so we may not investigate it at all.

3.2 Basic Table Compression

Oracle basic table compression achieves respectable compression ratios. Interestingly, Basic compression isn't compression at all, it is actually de-duplication at the block level [2]. To demonstrate de-duplication, take a look at the following example [2]:

Picture three rows in a block containing the following data:

- ('XXXX', 'abcdef', 254.32, 'CLOSED')

- ('XXXX', 'pqrstu', 17.12, 'CLOSED')
- ('AAAA', 'abcdef', 99.99, 'CLOSED')

Oracle could notice that the value 'XXXX' appears twice, that the value 'abcdef' appears twice, and that the value 'CLOSED' appears three times. Therefore, Oracle basic compression can create a table of repeated values in the block, and insert tokens into the rows to make them shorter. Our block would then look like this:

- T1 ('XXXX')
- T2 ('abcdef')
- T3 ('CLOSED')
- (T1, T2, 254.32, T3)
- (T1, 'pqrstu', 17.12, T3)
- ('AAAA', T2, 99.99, T3)

Furthermore, Oracle can rearrange the column order for each individual block to maximize the possibility of multiple columns turning into a single token. Notice that token T1 and token T3 both appear in all three rows; Oracle can rearrange the order that the columns are stored in this block to put those tokens side by side, and create a new token that represents the combination of the two individual tokens. Our block now becomes:

- T1 ('XXXX', T2) [a token made from a value and a token]
- T2 ('CLOSED')
- T3 ('abcdef')
- (T1, T3, 254.32) [notice how this row is now only 3 "columns"]
- (T1, 'pqrstu', 17.12) [also 3 columns]
- ('AAAA', T2, T3, 99.99)

Current project requirements include analyzing the way in which columns are organized and how blocks work in order to understand what Basic Table Compression actually does. Basic Table Compression is a feature of Oracle Database 12c Enterprise Edition [3], which the team uses. There is another compression method to be analyzed - Advanced Row Compression.

3.3 Advanced Row Compression

Oracle is a pioneer in database compression [3]. Advanced Row Compression was introduced in 2007, and maintains compression during all types of data manipulation operations, including conventional DML such as INSERT and UPDATE [3]. Additionally, Advanced Row Compression minimizes the overhead of write operations on compressed data, making it suitable for transactional / OLTP environments as well as Data Warehouses. The algorithm works "by eliminating duplicate values within a database block, even across multiple columns [3]. This is one of the main differences between Advanced Row Compression and Basic Table Compression: Basic Table Compression reduces redundancies with respect to rows at the block level, while Advanced Row Compression reduces redundancies in rows AND columns.

The compression ratio achieved in a given environment depends on the data being compressed, specifically the cardinality of the data [3]. Oracle claims that, in general, organizations can expect to reduce their storage spaces consumption by a factor of 2x to 4x" using Advanced Row Compression [3]. Furthermore, a significant advantage is Oracle's ability to read compressed blocks (data and indexes) directly in memory without uncompressing the blocks [3]. Aligning with current project goals, this method improves performance due to the reduction in I/O, and the reduction in system calls related to the I/O operations. Additionally, the buffer cache maximizes efficiency by storing more data without having to add memory [3].

3.4 Advanced Network Compression

Advanced Network Compression can be used to compress network data to be transmitted at the sending side, and then uncompress it at the receiving side to reduce network traffic [3]. Implementing Network Compression increases effective network throughput. Transmitting more data in less time increases SQL query response time [3], a huge benefit for large queries. Furthermore, it will save bandwidth by reducing the data to be transmitted, allowing other applications to use the freed-up bandwidth. On narrow bandwidth connections, with faster CPU, it could significantly improve performance.

3.5 Conclusion

Provided HP's division PageWide Web Press may be classified as Data Warehouse, Advanced Row Compression is critical in effectively reducing data stored in the database. As stated previously, Basic Table Compression is a feature of Oracle Database [3], which will help maintain the project budget of $0. The client already uses Advanced Row Compression, so despite its cost, the current project will not incur any additional charges. Analyzing both of the above compression methods is arguably the most important aspect of the project; discovering the way in which these compression methods work allows the team to make logically sound decisions in inserting, updating, retrieving, and managing data. If the client is satisfied that the team has accomplished, Advanced Network Compression will be considered as a stretch goal.

4 Performance Analysis

4.1 Introduction

The client aims to discover the fundamental mechanics of compression techniques and compare them against query performance. After analyzing the way in which compression methods work, performance of these methods must be measured with respect to CPU usage, Disk I/O, Memory, and Time. Toolkits are used to assist a DBA with analyzing the performance of queries made to a database, and there are multiple options ranging from PL/SQL queries run directly in the database to web applications. Efficiency is paramount, so the following database query analysis tools will be evaluated under consideration of the current scope and what is necessary to accomplish the task at hand: Oracle Enterprise Manager, SQLd360, and SQL Tuning Advisor. Special consideration will be made on creating our own performance analysis tool in order to ensure we obtain only the most essential metrics.

4.2 Available Tools

4.2.1 Oracle Enterprise Manager

Oracle Enterprise Manager (EM), like Basic Table Compression, is already included in Oracle Database 12c. It provides a convenient method for managing all Oracle deployments as well as a market-leading management and automation support for Oracle databases [4]. Oracle claims that EM increases DBA productivity by 80% and reduces database testing time by 90% [5]. Features include the ability to manage and maintain multiple databases from the web application, automating tasks, testing changes before pushing to production, diagnosing performance problems, and automated database tuning [6]. A significantly useful feature for analyzing query performance is Active Session History (ASH), which runs in the background and samples each active session once per second. Each sample provides detailed information about resources used by the session, which is useful for identifying performance issues.

4.2.2 SQLd360

SQLd360 is a free tool created by Mauro Pagano designed to provide a 360-degree overview around a SQL statement; it analyzes performance metrics of SQL queries and outputs a single zip file that allows off-line analysis [7]. The tool does not require any installation and can be executed by any user that has access to dictionary views: DBAs, developers, and sysadmins alike may use it [7]. Compared to EM, it only analyzes a single query and does not perform database optimization. SQLd30 extracts information from both AWR (licensed by Oracle under the Diagnostic Pack) and SQL Monitoring repository (part of the Oracle tuning Pack) [7], so although SQLd360 is free, it depends on using additional Oracle management packs, which are not. If users are not licensed with Oracle Diagnostics Pack or Oracle Tuning Pack, output content and value is substantially reduced.

4.2.3 SQL Tuning Advisor

SQL Tuning Advisor is a component of Oracle SQL Developer and analyzes SQL statements and offers tuning recommendations [8]. Because we are using SQL Developer, this may be a good option because it is already included. Unlike SQLd360, which only analyzes a single query, SQL Tuning Advisor "takes one or more SQL statements as an input and invokes the Automatic Tuning Optimizer to perform SQL tuning on the statements" [8]. "Oracle Database can automatically tune SQL statements by identifying problematic SQL statements and implementing tuning recommendations" [8].

4.3 Conclusion

Although Oracle Enterprise Management has many useful features and is capable of automatically optimizing SQL queries and database parameters, we will likely create our own analysis tool. SQLd360 is not the best option because it has multiple limitations - it may only analyze a single SQL query, and it relies heavily on Oracle Diagnostics Pack and Oracle Tuning Pack. Creating our own performance analysis tool allows the team to cherry pick metrics, reducing clutter and time spent becoming familiar with a new tool. Yes, time will be used to design a toolkit, but it makes sense to tailor a toolkit for the current scope. Creating our own analysis tool will allow a deeper understanding of database internals, which will ultimately lead to a more thorough and accurate analysis.

5 Summary

The team will use the following article and two technologies, which will aid in successful completion of the task at hand: Compression in Oracle - Part 1: Basic Table Compression, Compression techniques (Basic Table Compression and Advanced Row Compression), and our own Performance Analysis tool. There are no viable alternatives to the article Jonathan Lewis wrote - there is simply no dismissing the importance of the knowledge we acquired by reading it. The project requires use and analysis of both compression techniques, and, if the current task is completed ahead of schedule, a potential stretch goal is Advanced Network Compression. Researching and understanding database internals, while creating our own performance analysis toolkit, will allow us to efficiently measure only what is necessary.

6 References

[1] https://www.red-gate.com/simple-talk/author/jonathan-lewis/
[2] https://www.red-gate.com/simple-talk/sql/oracle/compression-oracle-basic-table-compression/
[3] http://www.oracle.com/technetwork/database/options/compression/advanced-compression-wp-12c-1896128.pdf
[4] http://www.oracle.com/technetwork/oem/enterprise-manager/overview/index.html
[5] http://www.oracle.com/technetwork/oem/db-mgmt/index.html
[6] http://www.oracle.com/technetwork/database/manageability/database-manageability-wp-12c-1964677.pdf
[7] https://mauro-pagano.com/2015/02/16/sqld360-sql-diagnostics-collection-made-faster/
[8] http://www.oracle.com/webfolder/technetwork/tutorials/obe/db/sqldev/r30/TuningAdvisor/TuningAdvisor.htm