A Compression Algorithm for Optimization of Storage Consumption of Non Oracle Database

Abha Tamrakar, Vinti Nanda

Abstract—Relational database are very important in satisfying today's information needs. This paper aims at optimization of storage consumption of non oracle database with the table compression algorithm of oracle 11g. Many concepts have been developed for compressing of relational database. But no concepts ever talked about the technique of compression of non oracle database with in the oracle environment. This paper discusses the use of OLTP Table compression algorithm given by ORACLE 11g which optimizes the storage consumption. But the challenges are faced by organizations when running several different databases and utilizing the heterogeneous services of the same. Data that are stored in non oracle system cannot be compressed by OLTP table compression algorithm given by Oracle 11g which reduces its size approximately about 70% ,so instead of migrating database system and loading the content of non oracle database system like SQL SERVER in ORACLE DATABASE format which is totally a wastage of time one can use oracle transparent gateway which is based on heterogeneous services technology. Oracle Transparent Gateways provides the ability to transparently access data residing in a non-oracle system from an oracle environment. After configuring the non oracle database in oracle environment and the relational database can be compressed by the table compression algorithm of oracle 11g.

Index Terms—Oltp Table Compression Algorithm, Oracle 11g, Oracle Transparent Gateway, Sql Server.

I. INTRODUCTION

The rapidly growing internet and related technologies has offered an unprecedented accessibility and redistribution of data. Relational database can be protected in several different ways. Like Oracle uses compression algorithm which provide security and optimizes storage space for its database. Heterogeneous data access is a problem that affects a lot of companies. A lot of companies run several different database systems. Each of these systems stores data and has a set of applications that run against it. Consolidation of this data in one database system is often hard in large part because many of the applications that run against one database may not have an equivalent that runs against another. Until such time as migration to one consolidated database system is made feasible, it is necessary for the various heterogeneous database systems to interoperate.

Oracle Transparent Gateways provide the ability to transparently access data residing in a non-Oracle system from an Oracle environment. This transparency eliminates the need for application developers to customize their applications to access data from different non-Oracle systems, thus decreasing development efforts and increasing the mobility of the application. Applications can be developed using a consistent Oracle interface for both Oracle and Microsoft SQL Server.

As a database administrator with multiple databases to manage, one should have the responsibility to make information available when and where it is needed and use the services provided by ORACLE in the non oracle database system like SQL SERVER

As organizations expand, it becomes increasingly important for them to be able to share information among multiple databases and applications. Data replication and integration enables us to access information when and where it is needed in a distributed environment. Oracle Database provides secure and standard mechanisms that enable communication between databases, applications, and users. These mechanisms include queues, data replication, messaging, and distributed access in both homogeneous and heterogeneous environments. If these organizations might prefer to centralize this data, at least in the short term, it might not be possible. These organizations must have a method of accessing these distributed data sources as if they were a single, centralized database. Using distributed SQL, applications and users can access and modify information at multiple Oracle or non-Oracle databases as if it resided in a single Oracle database. Because information does not need to be moved or copied, using distributed SQL to federate their distributed data sources provides organizations with the fastest, and easiest, path to information integration. If information is later moved, then it is not necessary to rewrite an application. This is especially useful for organizations that

Manuscript received July, 2012.

Abha Tamrakar, Computer Science and Engineering, Chhattisgarh Swami Vivekananda Technical University, Raipur, India, 09993844951, (e-mail: tamrakar.abha@gmail.com).

***Vinti Nanda**, Computer Science and Engineering, Chhatrapati Shivaji I institute of Technology, Bhilai, India, 09907142531., (e-mail: vintinanda@csitdurg.in).*

.

are transitioning to a consolidated approach, but need a method for accessing the distributed data now.
For example, by using distributed SQL with the appropriate Oracle Database Gateway, applications can access legacy data immediately, without waiting until it can be imported into an Oracle Database. Distributed SQL is also useful to organizations that want to perform ad hoc queries or updates on infrequently accessed data that is more appropriately located elsewhere.
Information about accessing and modifying information in multiple databases can be using Distributed SQL
Distributed SQL enables applications and users to query or modify information in multiple databases with a single SQL statement. Because distributed SQL masks the physical location of your data, you can change the location of your data without changing our application. Distributed SQL includes the following: distributed queries (which access data) and distributed transactions (which modify data).
In distributed transactions, the two-phase commit mechanism guarantees the integrity of your data by ensuring that all statements in a transaction either commit or roll back as a unit at each database involved in the distributed transaction.
When an application or user tries to commit a distributed transaction, the database to which the application or user is connected is called the global coordinator. The global coordinator completes the two-phase commit by initiating the following phases:

Prepare Phase: The global coordinator asks the other databases involved in the distributed transaction to confirm that they can either commit or roll back the transaction, even if there is a failure. If any database cannot complete the prepare phase, then the transaction is rolled back.
Commit Phase: If all of the other databases inform the global coordinator that they are prepared, then the global coordinator commits the transaction and asks all of the other databases to commit the transaction.
Oracle Database Gateway enables Oracle databases to access and modify data in a number of non-Oracle databases, including Sybase, DB2, Informix, Microsoft SQL Server, Ingres, and Teradata databases. This access is completely transparent to the end user. That is, one can issue the same SQL statements regardless of whether we are accessing data in an Oracle database or a non-Oracle database.
The data that is stored in SQL SERVER database cannot be compressed by OLTP table compression algorithm of Oracle 11g but by using ORACLE TRANSPARENT GATEWAY it is possible to facilitate the feature oracle database to non oracle database like SQL SERVER.

II. Literature Survey

Lossless Semantic Compression for Relational Databases by Haiming Huang [5] Renmin University of China, Beijing, P.R.China, 1998 proposed a semantic compression technique that exploits frequent dependency patterns embedded in the relational table. One advantage of this approach is that compression/decompression is performed at the tuple-level, which is desirable for integrating the compression technique into database systems. They showed that it is hard to compute an optimal compression solution. Therefore, an iterative greedy compression framework is offered to solve this problem. This work primarily focuses on the underneath component of the compression framework, that is to efficiently find dependency patterns in relational data to optimize the compression ratio. The experimental results on several real-life datasets demonstrate the effectiveness of their approach, as well as the efficiency and scalability.

M. Attalla and S. Lonardi [4] proposed a system, in which a simple variation on the classic LZ-77 algorithm that allows one to hide, within the compressed document, enough information to warrant its authenticity and integrity. The design is based on the unpredictability of a certain class of pseudo-random number generators, in such a way that the hidden data cannot be retrieved in a reasonable amount of time by an attacker (unless the secret bit-string key is known (2003).

Automatic relational database compression scheme design based on swarm evolution by Tian-lei Hu, Gang Chen, Xiao-yan Li and Jin-xiang Dong[3] presented a model with novel techniques to integrate a rapidly convergent agent-based evolution framework, i.e. the SWAF (Swarm Algorithm Framework), into adaptive attribute compression for relational database. The model evolutionally consults statistics of CPU load and IO bandwidth to select compression schemas considering both aspects of the trade-off. They have implemented a prototype model on Oscar RDBMS with experiments highlighting the correctness and efficiency of our techniques.

High Performance SQL Queries on Compressed Relational Database by Tian-lei Hu, Gang Chen, Xiao-yan Li and Jin-xiang Dong, [2]. They have developed a disk-based compression architecture, called DHIBASE, to support large database and at the same time, perform high performance SQL queries on single or multiple tables in compressed form. They have compared their system with widely used Microsoft SQL Server. Their system performs significantly better than SQL Server in terms of storage requirement and query response time. DHIBASE requires 10 to 15 times less space and for some operation it is 18 to 22 times faster. As the system is column oriented, schema evolution is easy.

But the above proposed method does not provide as much space consumption as that of oracle 11g table compression algorithm. So our paper uses the particular algorithm which provides more than 50 % of storage savings.

III. OUR APPROACH

In our approach we have used oracle transparent gateway to make non oracle database like SQL server database compatible with oracle and hence reduce the size of it by using OLTP table compression algorithm of Oracle 11g.

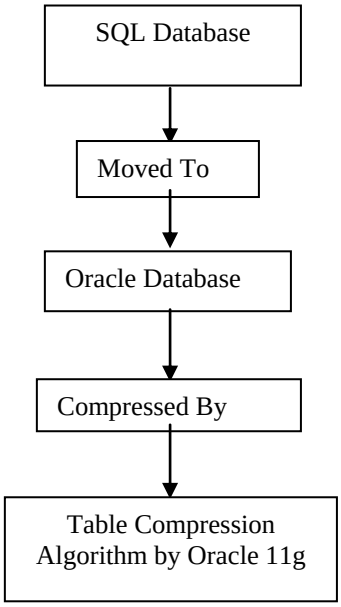

Fig. 1 Block Diagram of the proposed system

The following steps are involved in our approach:-
a. Compatibility of non oracle database system to oracle environment
b. Compression: The OLTP table compression algorithm given by Oracle 11g is for compressing the relational database table of sql server.

A. Algorithm

1. Create a relational table in the non oracle database like SQL server.
2. Make it compatible with oracle through Oracle Transparent Gateway;
   2.1 Configure the listener.
   2.2 Check the status of listener; if it's successful then go for the next step else diagnose the problem
   2.3 Start the listener.
   2.4 If the listener is started successfully then create a link between non oracle database and oracle.
   2.5 Bring the non oracle database to oracle
3. Compress the non oracle database with the table compression algorithm of oracle 11g.

B. Compression

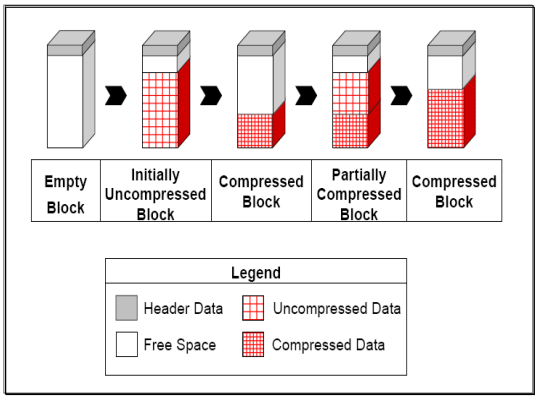

Fig 2.Steps Involved in OLTP Table Compression Process [1]

C. The advantages of OLTP Table Compression Algorithm are:

•Structured/Relational data Compression.
•Unstructured data compression
•Compression for backup data
•Network transport compression
•Reduces resource requirements and costs.
•Storage System
•Network Bandwidth
•Memory Usage

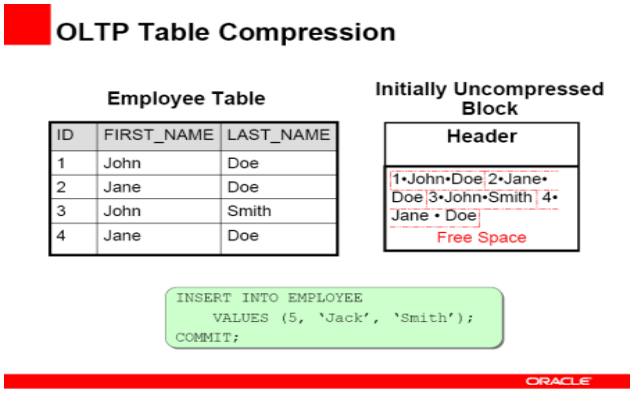

Fig 3.OLTP Table Compression Technique[1]

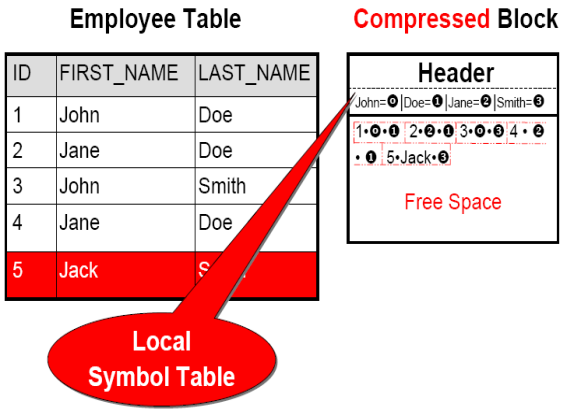

Fig 4. Free Space available after applying OLTP Table Compression Algorithm [1]

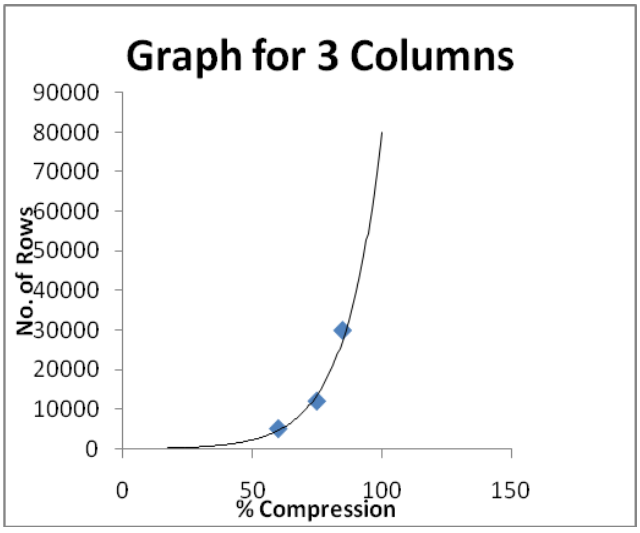

Fig 5. Compression for Tables having 3 columns

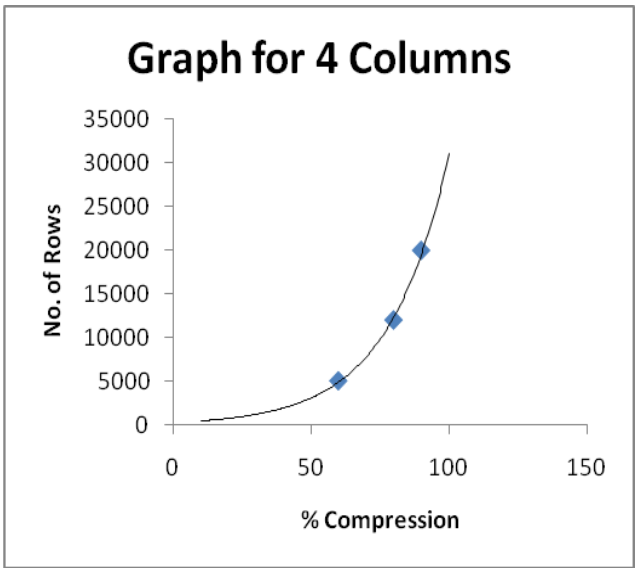

Fig 6. Compression for Tables having 4 columns

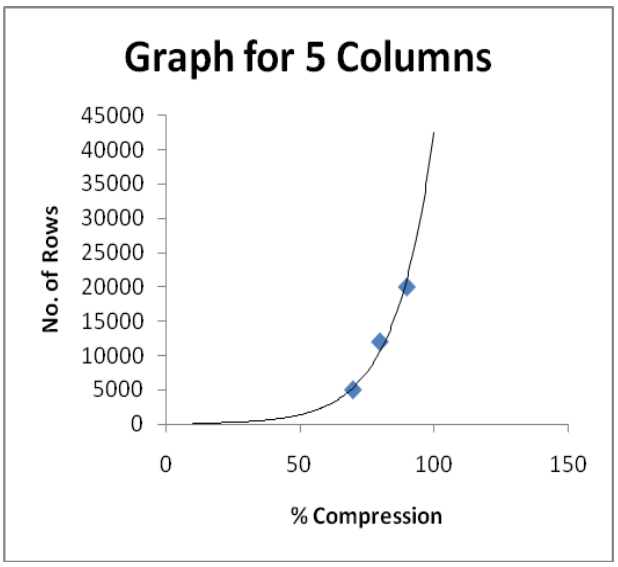

Fig 7. Compression for Tables having 5 columns

The following three graphs show the percentage of compression for different number of rows with tables having column 3, 4 and, 5. The figure shows the percentage of compression increases with the increase in the rows of table more effectively as compared to the number of columns.

IV. CONCLUSION

The compression technique used in our approach compresses relational database of non oracle database like SQL Server through the OLTP Table compression algorithm of Oracle 11g using Oracle Transparent Gateway. Since, we are using Table compression algorithm given by oracle 11g but the data is in SQL Server we have to make it compatible with the oracle system which is done through oracle transparent gateway then we can use the compression algorithm of oracle.

ACKNOWLEDGMENT

I would like to thank Ms. Vinti Nanda, Asst. Professor in CSIT, Durg (C.G) for her cooperation, guidance to make us enthusiastic. Secondly I would also like to express my sincere thanks to Mrs. Tripti Sharma, Head of department of Computing Science and Engineering, CSIT, Durg (C.G) for her continuous efforts in encouraging and guiding us to become successful engineers.

REFERENCES

[1]http://www.oracle.com/technetwork/database/focus-areas/storage/advanced-compression-whitepaper-130502.pdf

[2]. Mohammad Masumuzzaman Bhuiyan, Abu Sayed Md. Latiful Hoque, High Performance SQL Queries on Compressed Relational Database, JOURNAL OF COMPUTERS, VOL. 4, NO. 12, DECEMBER 2009.

[3].Tian-lei Hu, Gang Chen, Xiao-yan Li and Jin-xiang Dong, Automatic relational database compression scheme design based on swarm evolution, Journal of Zhejiang University - Science A Volume 7, Number 10 (2006), 1642-1651

[4] M. Attalla and S. Lonardi. "Authentication of LZ-77 Compressed Data." In Proceedings of the ACM Symposium on Applied Computing, Florid, (2003)

[5]. Haiming Huang, Lossless Semantic Compression for Relational Databases, B.E., Renmin University of China, Beijing, P.R.China, 1998.

ABHA TAMRAKAR: I received my B.E in Computer Science and Engineering from Shri Shankaracharya College of Engineering and

Technology of Chhattisgarh State, India in 2009. I am now Pursuing my M.Tech at Chhattrapati Shivaji Institute of Technology State, India. My area of interest includes compression, security, image processing**.**

VINTI NANDA: She received her M-TECH in Computer Science and Engineering from Rungta College of Engineering and Technology of Chhattisgarh State, India in 2010.She received her B.E in Computer Science and Engineering from Shri Shankaracharya College of Engineering and Technology of Chhattisgarh State, India in 2006.She is working as Asst. Professor in Chhattrapati Shivaji Institute of Technology(CSIT), DURG(C.G)