

Oracle Application with Database Testing tool

Navyashree. B S¹, Dr. S Kuzhalvaimozhi²

¹Department of Information Science and Engineering, The National Institute of Engineering, Mysuru,
Karnataka, India

ABSTRACT

Oracle Database is simple and highly scalable database. It is majorly used for online transactions processing. Large enterprises widely use Oracle Database in their organization in multiple departments like HR Department, IT Department etc., Each department would host their database with large schema and tablespace. Ensuring database is sized well to handle multiple users to get optimal performance is very important. Oracle deployment can be done with single node deployment and Real Application Cluster (RAC) with multiple nodes. Database testing tools help in measuring the workloads on the database. The tool will provide measured values of latency, throughput and Input / Output Operations per Second (IOPS). Mainly there are two notable Oracle Database tools like Vdbench testing tool and Silly Little Oracle Benchmark (SLOB) testing tool.

Keywords: Oracle Database, Real Application Cluster, Vdbench, Silly Little Oracle Benchmark.

I. INTRODUCTION

A database is a collection of data and a database server is a vehicle for information management. The primary purpose of a database is to store and retrieve related information. These servers can manage large amounts of data, reliably in a multiuser environment. At the same time, they also prevent unauthorized access and provide efficient solutions for failure recovery. A database's physical and logical structures are separate. The physical storage of data can be managed independently of logical storage structures.

Oracle Database is a Relational Database Management System (RDBMS) from the Oracle Corporation. It is the first database designed for enterprise grid computing and flexibility in managing information and application in a cost-effective manner. Large pools of industry-standard modular storage and servers are created by enterprise grid computing. With Oracle Database architecture,

each new system can be rapidly provisioned from a pool of components. Extra hardware to support peak workloads is not required since capacity can be easily added or reallocated from the resource pools as needed.

Users may directly access data objects through structured query language. This is a feature of the Oracle Database's relational database framework. Global enterprises which handle data across WANs often use Oracle Database because of its scalability and robustness. It has its own network component to allow communications across networks. It is supported on most major platforms including Windows, UNIX, Linux and Mac OS. Oracle Database deployments are popular in RHEL.

Figure 1 Represents Oracle Database Architecture. Oracle Database server consists of an Oracle database and an Oracle instance.

Oracle Database have three different type of files, Datafiles, Control files and Redo Log files. Datafile stores all data related to database and it has logical storage structure called tablespace. Control files store the physical structure of the data in database, Redo Log files store the redo logs in the database and used for database recovery process in case of database crash. Oracle instance is the combination of System Global Area(SGA), Program Global Area(PGA) and Background processes.

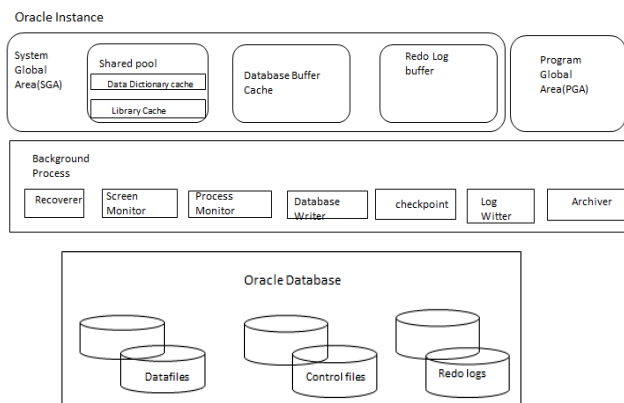


Figure 1. Oracle Database Architecture.

System Global Area starts one or more Oracle process by allocating the memory space in the database. It consists of Shared Pool Database Buffer Cache, Redo Log Buffer. Shared Pool Database plays an important role in the SGA, it is the temporary place to store the information in the RAM heap and used to execute the sequential query language statements. Program Global Area is the memory place to store the data information and control information for a server process. Background Processes includes Recoverer, System Monitor, Process Monitor, Database Writer, Checkpoint, Log writer and Archiver.

II. LITERATURE SURVEY

Most of the business companies and institutions are facing problem in maintaining the huge related information and processing it properly. The process of sorting the useful information leads to both individual and organizational challenge. This paper provides the model to classify the information using the rule-based classification by loading the document

in the Oracle12c database. Oracle 12c provides a stable foundation base. It has a very close integration between the Oracle SQL execution engine and the query plan optimizer. A single storage can be used by the structured data and text data. Easy in maintenance operations such as data backup, tuning and other database tasks. This makes the storage mechanism more efficient for queries [1].

The storage of critical information in the companies or in the organizations is highly challenging. The information is not only in structured text form or numeric data. unstructured content such as images, videos, PDF, word documents and other types are becoming more and more common in various fields. This paper presents the model for storing unstructured data such as documents in Oracle 12c database using basic file and secure file method. Secure file method helps in storing the critical data in different forms like videos, PDF and word documents [2].

The three artifacts are generated to reduce the problem in the automated approach for the systematic black-box testing of database management using the relational constraint they are

- 1) Using SQL queries for testing
- 2) Proper input data to populate test databases
- 3) Expected results for the SQL queries used for testing.

This paper presents a detailed description of the framework for automated SQL Generation using the Alloy tool-set and the experimental results of testing database engines using our framework and explains the SQL grammar constraints can be solved by translating them to Alloy constraints to provide semantically and syntactically correct SQL queries [3].

Oracle Real Application cluster(RAC) allows two or more database instances run on different server nodes in a cluster.

It is an instance recovery, when a single database instance is suffering from the hardware or the software failure it will not affect the other nodes which are present in the cluster and still the cluster is able to provide the database service. During the instance recovery, database resources which need recovery must be identified and should be locked then repaired. The client's requests are blocked for the database access at this time. The complete database is in a frozen state called brownout.

This paper describes the uses of Buddy instance, a mechanism which reduces the brown-out time and the duration of the instance recovery. It is introduced in the 12.2.0.1 release of Oracle Database to reduce the period of identification of resources needing recovery [4].

III. ORACLE DEPLOYMENT

Oracle Deployment can be taken place in two different ways.

- Oracle Database Deployment in Single Node.
- Oracle Database Deployment in Real Application Cluster with Multi Node.

A. Oracle Database single node deployment.

Oracle Single Node Deployment is a single instance of an Oracle Real Application Cluster. It enables the database to run on one node in the cluster in all normal operations. Single node deployment method adds flexibility and reduces management overhead by providing the standard deployment for Oracle Database in the enterprise. Figure 2 represents the Single Node Deployment in the Oracle database which contains only a single database instance.

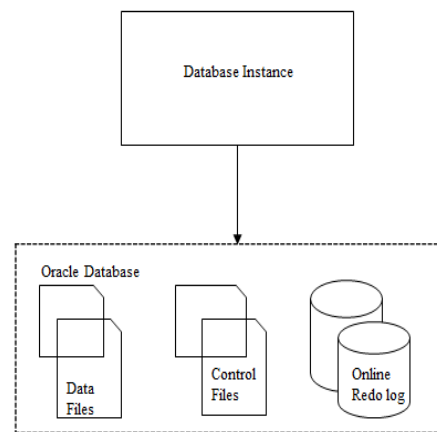


Figure 2. Oracle Single Node Deployment.

It utilizes a shared disk architecture to provide high availability; the volume management and file system used for storing database data must be cluster-aware. It provides an Oracle Database failover solution.

Oracle Automatic storage management is the recommended clustered volume manager for the Oracle database. Oracle Grid Infrastructure is required for the deployment of the Oracle database. Oracle Database should contain the similar hardware setup as an Oracle real application cluster database and be certified on Oracle Virtual Machine. Combining Oracle single node and Oracle virtual machine increases the benefits of Oracle Virtual Machine with the high availability and scalability of the Oracle Database.

Eliminates the single Database server as a single point of failure and takes further advantage of clustering to apply rolling patches and database service relocation without incurring downtime. Provides all the software components required to easily deploy Oracle database instances on a pool of servers and take full advantage of the performance, scalability and availability that clustering enables. Multiple single database instances can be configured on a physical Oracle One Node server. In addition, with full support for Oracle VM, multiple database instances can be configured in a virtualized environment.

B. Oracle Real Application Clusters with multiple node deployment.

Oracle Real Application Clusters allows two or more computers to run Oracle RDBMS software parallelly while accessing a single Database, which provides the clustering. In an Oracle Real Application Clusters environment two or more instances simultaneously access a single database. The main objective of Oracle Real Application Clusters is to provide performance, scalability, resilience and high availability of data at multiple instance level. This allows an application or user to connect to either computer and have access to a single similar set of data. Multiple instances which are inter connected can access the database at a single point in time.

The Architecture of Oracle Database is differing from the Oracle Databases single-instance, which contain two or more instances are connected to single Oracle Database containing the different types of files like Datafiles, Control files, Redo log files. Each Oracle RAC database instance also has: One additional thread of redo for each instance are present. An instance-specific undo tablespace is present. Figure 3 Represents the Oracle RAC Database Architecture. shared cache architecture is present in the clustered architecture of real application cluster which overcomes the limitations of traditional shared-nothing and shared-disk approaches to provide highly scalable and available database solutions for all business applications. It is the major component of Oracle's private cloud architecture. It provides customers with the highest database availability by removing individual database servers as a single point of failure. In a clustered server environment, the database itself is shared across a pool of servers, if any server in the server pool fails, the database continues to run on surviving servers.

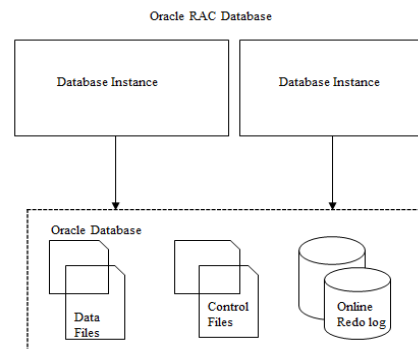


Figure 3: Real Application Cluster Architecture.

On Oracle Real Application Cluster not only enables customers to continue processing database workloads in the event of a server failure, it also helps in further reduce costs of downtime by reducing the amount of time databases are taken offline for planned maintenance operations.

It enables the deployment of Oracle Databases across a pool of clustered servers. This enables customers to easily re-deploy their single server Oracle Database onto a cluster of database servers, and thereby take full advantage of the combined memory capacity and processing power the Clustered database servers provide. It provides all the software components required to easily deploy Oracle Databases on a pool of servers and take advantages of the performance, scalability and availability that clustering provides. It utilizes Oracle Grid Infrastructure as the foundation for Oracle Real Application Clusters database systems. Oracle Grid Infrastructure includes Oracle Cluster ware and Oracle Automatic Storage Management (ASM) that enable efficient sharing of server and storage resources in a highly available and scalable database cloud environment.

ORACLE DATABASE TESTING TOOL.

A. SILLY LITTLE ORACLE BENCHMARK

Silly Little Oracle Benchmark(SLOB) is one of the useful I/O workload generator for the Oracle Database by using the minimal CPU resource. It is the best Input Output workload generation toolkit for the database. It performs the sustained throughput test by considering a large transactional

workload is driven through the database over a long period to measure the ability of the storage system to offer a sustained and predictable response. It performs the sequential query language execution in the Oracle Database with the minimal host CPU utilization.

Oracle Database is highly CPU-intensive in the input output operations, has much code associated with resources sharing, resource sharing is for the mutual exclusion in the database operation.

SLOB minimize this overhead by avoiding contention on the shared resources.

SLOB requirements for Oracle Database testing

- An Oracle installation on the server where SLOB is installed, with a working SQL*Plus
- An Oracle database.
- SYSDBA access to the database.

SLOB configuration file should comprise the parameters like

UPDATE_PCT, RUN_TIME, WORK_LOOP, WORK_UNIT, SCALE, REDO_STRESS, LOAD_PARALLEL_DEGREE, SHARED_DATA_MODULUS. Fig represents the SLOB config file with parameters assigned value.

```
UPDATE_PCT=25
RUN_TIME=28800
WORK_LOOP=0
SCALE=1398101
WORK_UNIT=3
REDO_STRESS=LITE
LOAD_PARALLEL_DEGREE=
16
```

Figure 4. Slob configuration file

UPDATE_PCT

Set to X where X is the percentage of all SQL that will be UPDATE DML executions. Setting N to value 0 or to value 100 are the functional equivalent of prior generations of SLOB where one might either

execute 100% SELECT or 100% UPDATE workload. Values between 51 and 99 are non-deterministic.

RUN_TIME

Set to X where X is the number of seconds you want the test to run. This can be overridden with WORK_LOOP. If the value set, RUN_TIME the value should set WORK_LOOP to 0.

WORK_LOOP

Set to X where X is the fixed number of loop iterations. Testing in this manner is a way to measure job completion as opposed to fixed test period with sampling of stats. If you want to do a fixed-iteration test, recommend value to RUN_TIME large enough to make sure the run doesn't end based on RUN_TIME.

SCALE

Set to X where X=10000 is the old SLOB scale. X=250000 is ~256GB with 8KB block and 128 schema users. WORK_UNIT SLOB picks a random 256 blocks to work on in each iteration of the work loop.

REDO_STRESS

Set either to HEAVY or any other non-null value. HEAVY maps to the old writer. SQL heavy. Any value other than HEAVY reduces the redo payload in MB/s by about 80%.

SHARED_DATA_MODULUS

This parameter controls the degree of shared data contention. If set to a non-zero value it is used in modulo arithmetic for every UPDATE to direct either to the session's private schema or shared data manipulation.

LOAD_PARALLEL_DEGREE

The setup.sh script uses this to control concurrent data loading, 1 per core at a minimum and 1 per processor thread at a maximum.

SLOB runs in the Oracle database to calculate the input output operations per second, latency and throughput. By plotting the graphs with respect to the number of users reads the Oracle database. Figure 4 represents the graph of IOPS, Figure 5 represents the graph of Latency and Figure 6 represents the graph of throughput.

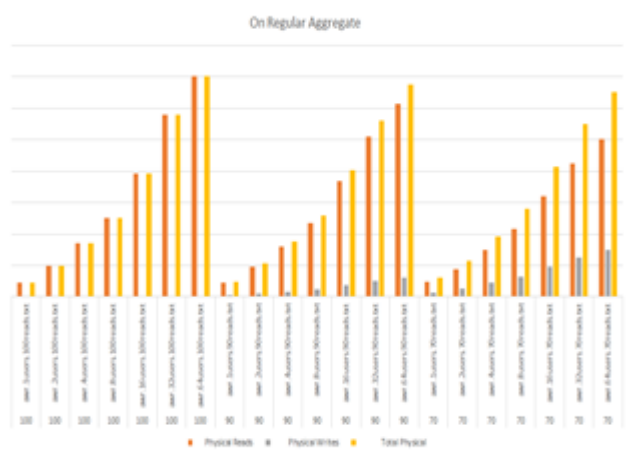


Figure 4, Graph with respect to IOPS in Oracle database.

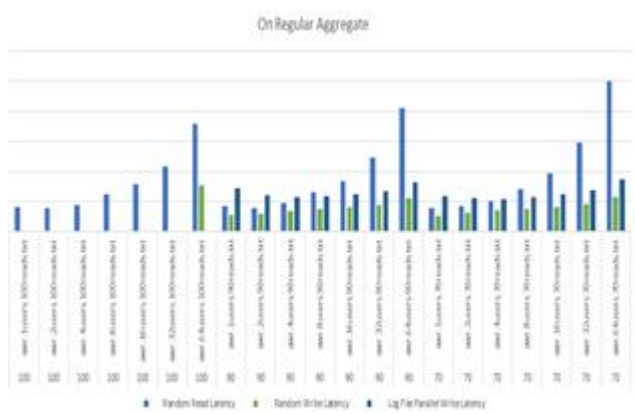


Figure 5. Graph with respect to latency in Oracle database.

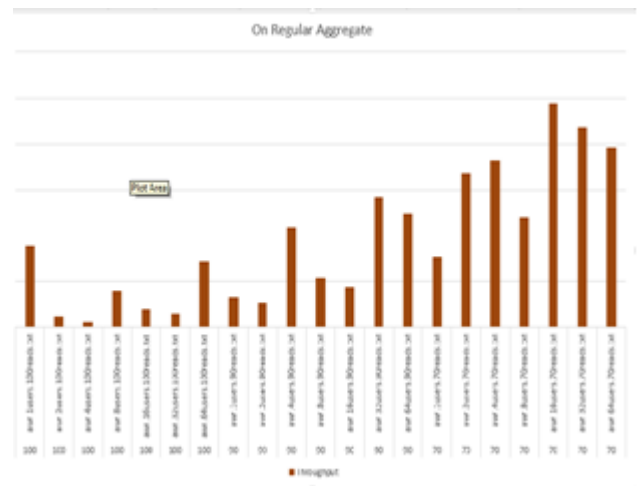


Figure 6. Graph with respect to Throughput in Oracle Database.

IV.CONCLUSION

Database application like Oracle can grow large with multiple databases over period. As with data growth, data management becomes very challenging. Testing tools help in measuring the workloads on the database. The tool will provide measured values of latency, throughput and Input / Output Operations per Second. The notable Oracle Database tool is Silly Little Oracle Benchmark (SLOB) testing tool helps in providing the measured values and benchmarking for existing and future storage products and databases. It supports logical read scaling, testing physical random single-block reads and writes. Silly Little Oracle Benchmark(SLOB) is an I/O workload generator for Oracle Database that drives lots of real database I/O using minimal CPU resources.

V.REFERENCES

1. Hendro Nindito, Andreas Raharto Condrobimo, Albert Versius Dian Sano,” Rule Based Classification for Diseases Articles Using Oracle Database 12c”.
2. Shadi Abdul Khalek, Sarfraz Khurshid,” Systematic Testing of Database Engines Using a Relational Constraint Solver 2011 Fourth IEEE International Conference on Software Testing, Verification and Validation

3. Oracle, "Unstructured Data Management with Oracle Database 12 c," White Pap., no. September 2014.
4. K Gayathri and M. Chitra, "Review of Classification Techniques for Identifying Disease Cause and Diagnosis," *Int. J. Adv. Res. Comput. Sci. Software Eng.*, vol. 4, no. 11, pp. 521–524, 2014
5. G Mamakis, A. G. Malamos, J. A. Ware, and I. Karelli, "Document Classification in Summarization," *J. Inf. Comput. Sci.*, vol. 7, no. 1, pp. 25–36, 2012.
6. Raquel Blanco, Javier Tuya, Ruben V. Seco, "Test Adequacy Evaluation for the User-database Interaction: A Specification-Based Approach", *Software Testing Verification and Validation (ICST) 2012 IEEE Fifth International Conference on*, pp. 71-80, 2012.
7. Oracle, "Oracle Text, An Oracle Technical White Paper," Oracle Tech. White Pap., no. June, 2013.
8. K Gayathri and M. Chitra, "Review of Classification Techniques for Identifying Disease Cause and Diagnosis," *Int. J. Adv. Res. Comput. Sci. Software Eng.*, vol. 4, no. 11, pp. 521–524, 2014.
9. Shadi A. Khalek, Bassem Elkarablieh, Y. O. Laleye, and Sarfraz Khurshid. Query-Aware Test Generation Using a Relational Constraint Solver. In *ASE '08*, pages 238-247, IEEE
10. Nicolas Bruno, Surajit Chaudhuri, and Dilys Thomas. Generating queries with cardinality constraints for DBMS testing. *IEEE Transactions on Knowledge and Data Engineering*, 18(12):1721–1725, 2006
11. Meikel Poess and Jr. John M. Stephens. Generating thousand benchmark queries in seconds. In *VLDB'2004*, pages 1045– 1053.
12. Nicolas Bruno and Surajit Chaudhuri. Flexible database generators. In *VLDB '05*, pages 1097– 1107. VLDB Endowment
13. Yuetang Deng, Phyllis Frankl, and David Chays. Testing database transactions with agenda. In *ICSE '05: Proceedings of the 27th international conference on Software engineering*, pages 78– 87, 2005.
14. Carsten Binnig, Donald Kossmann, Eric Lo, and M. Tamer Ozsu. Qagen: generating query-aware test databases. In *SIGMOD Conference*, pages 341–352, 2007.
15. MySQL opensource database. <http://www.mysql.com/>.