Load Balance Monitoring in Oracle RAC

Neha Chandrima
PG Student, Department of Computer Science & Engineering
LNCT Bhopal India-462022

Sunil Phulre
Asst. Professor, Department of Computer Science & Engineering
LNCT Bhopal India-462022

Vineet Richhariya, Ph.D
Professor& Head, Department of Computer Science & Engineering
LNCT Bhopal India-462022

ABSTRACT

Oracle RAC offers and provides very attractive and promising features for today's challenging market scenario, where safety and availability of database is the basic need and demand of clients who are working on mission critical databases. The two most alluring features which RAC provides are high availability and load balancing. Load balancing is performed by the load balancer which most of the time works as desired; but it is noticed that failure of load balancer could be a single point of failure for the entire RAC system. In the present research work our aim is to monitor the load balancing feature of Oracle RAC in order to keep a close watch on whether switchover between nodes is happening correctly and smoothly or not. The concept of relative entropy is taken as inspiration, which helps us compare the randomness of the nodes and also the entire RAC system. A script is designed to monitor the number of sessions in each node which should run at the expected peak hours. These peak hours are discovered by extensive monitoring of sessions in each node for 30 days continuously. A threshold limit for number of sessions in each respective node is to be defined in prior, which should be near about but less than maximum session limit defined for the load balancer. As soon as the number of sessions in a particular node reaches its threshold limit, an email is sent to the DBA team which in turn alerts the team to keep a close watch on the switchover.In the proposed scheme the motto is to provide the client with a better service with the help of proper monitoring. To insure an even workload distribution, a clustered database must employ methods to distribute incoming sessions evenly and effectively across the various components of its cluster. As soon as the DBA gets email showing the threshold limit is reached, the DBA must become alert. If he notices that switchover or in other words load balancing is not happening properly, he can do a manual switchover to prevent the client from suffering with rejection of server. In present work Relative Entropy calculation technique is used as an inspiration to calculate the randomness of the RAC system based on its load balancing feature. Proposed scheme can minimize the cost as we can tune up the memory or SGA/PGA based on our result reducing the time to fetch as well. The Relative Entropy calculation scheme of detecting the randomness of Oracle RAC system is a truly new approach based on which we can monitor as well as forecast the behavior of its load balancing act. Proposed method also could help to detect any upcoming issue related to load balancing which could interrupt the client services. The proposed mechanism has overcome some of the limitations like it can be automated without any prior installation.

General Terms

Oracle RAC database, Load balance monitoring of Oracle RAC database, Oracle Real Application clusters.

Keywords

Oracle RAC, Load Balancing, DBA, Relative Entropy, Clustered database

1. INTRODUCTION

Oracle RAC is an important clustered oracle databases. It uses oracle clusterware software for the infrastructure to bind multiple servers so that they operate as a single system. A cluster comprises of multiple interconnected computers or servers that appear to be one server to end users and applications [1]. It is Oracle RAC architecture option that provides a single system image for multiple servers to access one Oracle database. In Oracle RAC, each Oracle instance usually runs on a separate server. The combined processing power of the multiple servers can provide greater throughput and Oracle RAC scalability than is available from a single server. Figure 1 depicts line diagram of Oracle database with Oracle RAC architecture:

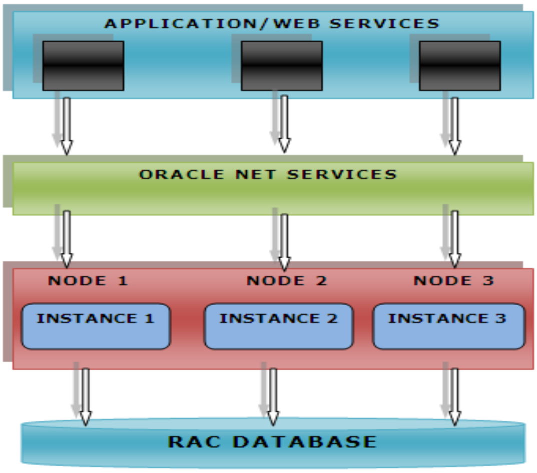

Figure 1: Oracle database with Oracle RAC architecture

Oracle RAC provides high availability and scalability for all application types. The Oracle RAC infrastructure is also a key component for implementing the Oracle enterprise grid computing architecture. Many research works have been carried out in the field of Oracle RAC which gave important findings. Some significant contributions of the researchers are highlighted here: Gong Weihua and Wang Yuanzhen [2] reported that in heterogeneous database cluster system, performance is very much close to the computing capabilities of nodes and their different categories of workloads. They introduced an innovative method to determine a load status of nodes by the weighted load values with consideration of utility of different resources and workload types in load balancer. Stephane Gancarski [3] Proposed a new innovative solution to balance load of autonomous applications and oracle databases. This proposed solution is very much same as distributed shared memory in that it also provides shared address space to application with distributed and replicated databases. Jin chen et al [4] devised a process which detects outlier for fine grained load balancing in database clusters. Their work is focused towards reduction of cost of ownership in large data centers and emphasized the need for database system method for automatic performance tuning and efficient resource usage. Automatic provisioning of database servers to applications and virtualization techniques, like, live virtual machine migration was proposed as useful tools to address this problem. Don MacVittie [5] reported that databases are essentially the heart of the IT infrastructure that today's business runs on .Load balancing is challenge for every organization. He suggested that as workloads continue to increase, organizations will use both load balancing and clustering databases to meet performance goals with commercial, off-the-shelf servers. F5 BIG-IP products help to improve the performance of database clusters by expanding Oracle FAN out to non–FAN enabled clients, thus offering fast connection resets. Gia- Khanh Nguyen and Tim Read [6] discussed about Real Application Clusters on oracle solaris zone clusters. They explained how to deploy, Oracle RAC on oracle solaris zone clusters. Deepali Kadam et al [7] worked on automatic failover and load balancing for Oracle real Application Clusters. RAC enables us to use clustered hardware by running multiple instances against the same database. The database files are stored on disks that are either physically or logically connected to each node, so that every active instance can read from or write to them. René Kundersma [8] reported upgrading to Oracle RAC and discussed about 11gR2 Oracle Real Application Clusters. It was stated that, Oracle Database 10g and Oracle Database 11g are supported with Oracle Clusterware 11g Release 2 Howard Karlof and Kenneth Shirley [9] reported the concept of maximum entropy summary trees used in databases. They suggested that the best choice of which summary tree to use among those with a fixed number of nodes is the one that maximizes the information-theoretic entropy of a natural probability distribution associated with the summary tree, and also provided a pseudo-polynomial-time dynamic-programming algorithm to compute this maximum entropy summary tree, when the weights are integral. John McHugh and Markus Michalewicz [10] discussed various Oracle RAC deployment scenarios and provides best practices for an optimized Oracle RAC deployment in Oracle VM environments for Oracle VM versions 2.1.2 through Oracle VM 3.0.3. version specific information is noted accordingly..

2. PROPOSED METHODOLOGY

In Oracle RAC databases, the service performance rules which is configured, controls the amount of work that Oracle database allocates to each available instance for that service. It is well known that Oracle RAC provides best available features for database. 'DATABASE' which most of the time is highly crucial information for a user; is required to be completely safe and should be available to the user whenever he tries to retrieve it. With this key concept, Oracle RAC is designed. The most attractive feature of Oracle RAC is high availability and load balancing between nodes. Load balancing is performed by the load balancer which most of the time works as desired; But it is noticed that failure of load balancer could be a single point of failure for the entire RAC system. In the present research work, the aim is to monitor the load balancing feature of Oracle RAC in order to keep a close watch on whether switchover between nodes is happening correctly and smoothly or not. The concept of relative entropy is taken as inspiration, which helps in comparing the randomness of the each node and also the entire RAC system. In this, A script is designed which runs at the time defined by DBA. This is usually, the peak hours when service request/ load is maximum. The script fetches number of sessions connected to all the respective RAC nodes. A threshold limit is predefined in the script. Once the RAC node reaches the threshold limit, an email alert is sent to the DBA team, which alerts the team to keep a close watch on the node switchover and if any problem is noticed in the switchover by the DBA , it could be done manually. This ensures that the system is uninterrupted even if the load balancer fails. The methodology devised for the work is given in Figure: 2**.**

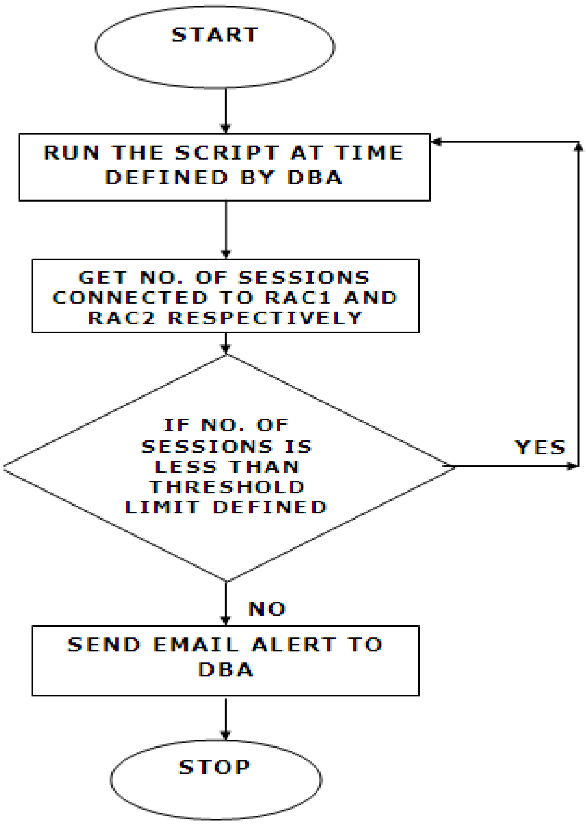
START
RUN THE SCRIPT AT TIME DEFINED BY DBA
GET NO. OF SESSIONS CONNECTED TO RAC1 AND RAC2 RESPECTIVELY
IF NO. OF SESSIONS IS LESS THAN THRESHOLD LIMIT DEFINED
YES
NO
SEND EMAIL ALERT TO DBA
STOP

Fig.2 Methodology of the Present Work

3. IMPLEMENTATION PROCEDURE

In present research work, the aim is to make the Oracle RAC system more strengthened by adding a feature which sends e-mail alerts to the database administrator when the number of connections is about to reach the maximum limit. This feature could be very useful for an Oracle RAC DBA as it gives prior hint to them that when to keep their eyes widely open and see whether the switchover is happening properly between the nodes or not.

The relative entropy based scheme could be used to plot the probabilistic graphs which may be used as reference in future. For an example if a relative entropy based scheme is used to plot a graph by taking number of connections at different point of time for all days of a week and continue doing this for several weeks, we can get a pattern which is followed by load balancer. Based on these patterns the limits could be defined for getting e-mail alerts. One such script is designed in the present research work, which sends email alerts to the DBA when the maximum number of connection defined is 500. As soon as the number of sessions reaches 475, the system would send an alert message to the DBA for keeping a close watch on the switchover act. Ideal behavior of node 1 and node 2 is depicted in Figure: 3 in the form of graph plotted between different time intervals and number of sessions.

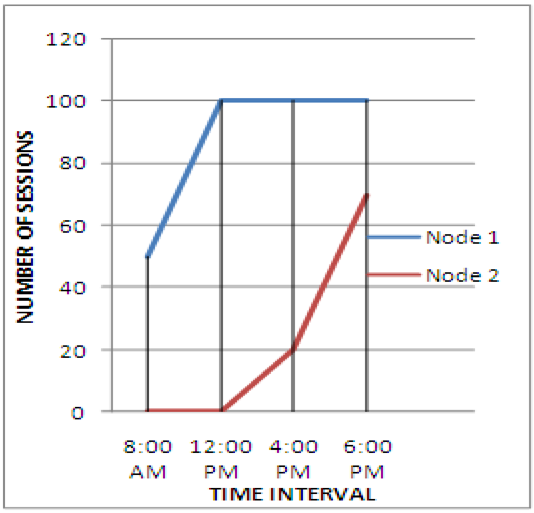

Figure 3: Graph showing ideal behavior of Node 1 and Node 2 at different time intervals

The above graph shows the behavior of Node 1 and Node 2 at different time intervals when the maximum number of connections defined is 100. All the connections are diverted to Node 1 till it reaches the maximum defined limit. As soon as the maximum limit is reached, the connections get diverted to Node 2. The job of diverting these connections is handled by the listener and load balancer. But sometimes the failure of load balancer could be a single point of failure for the entire Oracle RAC system. Thus it is very much required for keeping a check on the entire system time to time. The load balancer could be defined to direct the connections to the node which is least loaded or to the node which is providing best service for similar service request in the past.

4. RESULTS

On successful implementation of proposed methodology, the important findings of the work are drawn. These are given in the form of graphs and there interpretation is described here.

4.1 First Scenario

Figure 4 shows the ideal scenario of switchover with session limit 500.

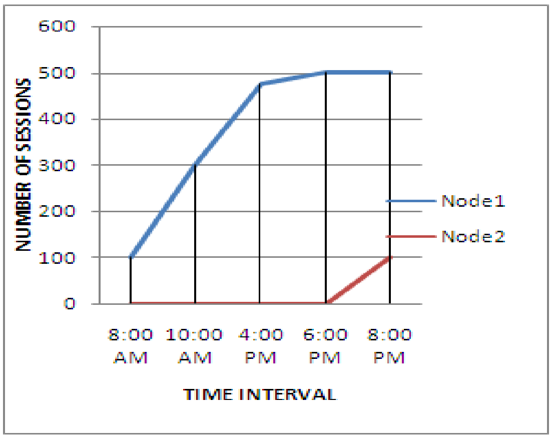

Figure: 4 Graph showing ideal scenario of switchover with session limit 500

The above graph is plotted between time interval in X axis and number of sessions in Y axis. The script was configured to run at peak hours. The below table is the tabular interpretation of graph in tabular form.

Table 1: Tabular interpretation of Figure: 4

TIME	NODE 1	NODE 2
8:00 AM	100	0
10:00 AM	300	0
4:00 PM	475	0
6:00 PM	500	0
8:00 PM	500	100

The above graph shows an ideal scenario of switchover where the maximum limit defined for sessions was 500. A threshold limit for number of sessions was defined at 475. As soon as it reaches 475, threshold is reached and an email alert is sent to the DBA team. This setup helps to keep an eye on the system if switchover is happening smoothly or not. In an ideal scenario switchover happens smoothly and the graph plotted will be something like Figure: 4. But in case the switchover doesn’t happen, the graph will be like Fig. 5, where the entire load is taken by Node 1 and Node 2 is sitting idle. In such case the RAC system may fail after sometime. As soon as the instance reaches its maximum limit (at instance level) or if it becomes too busy handling other requests, it will start rejecting new connections and the customer may have to suffer.In this research work the motto is to provide the client with a better service with the help of proper monitoring. As soon as the DBA gets email showing the threshold limit is reached, the DBA must become alert. And if he notices that load balancing is not happening properly, he can do a manual switchover to prevent the client from suffering with rejection of server.

4.2 Second Scenario

A scenario when switchover doesn’t happen at defined value (say, max limit 500):
The result is shown by plotting graph given in Fig.5.

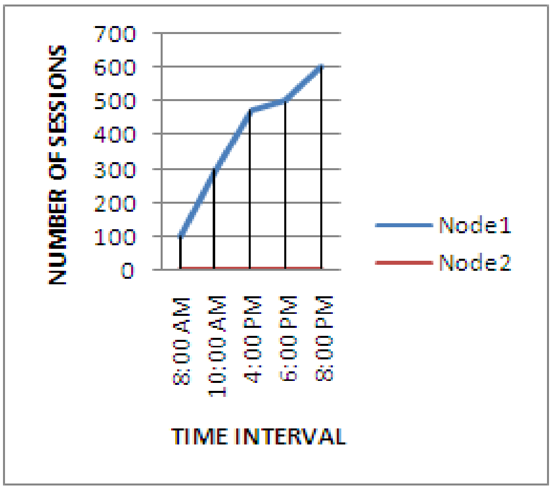

Figure: 5 Graph showing a scenario when switchover doesn't happen at max limit 500

The above graph shows a situation where switchover doesn't happen at the max limit. Node1 is taking all the load and Node 2 is sitting idle. The threshold was achieved when number of session was 475 and maximum limit of sessions defined was 500. But unfortunately, even after reaching maximum limit the connections were not diverted to Node 2. The below table shows the data interpretation in a tabular form.

Table 2: Tabular interpretation of Figure: 5

TIME	NODE 1	NODE 2
8:00 AM	100	0
10:00 AM	300	0
4:00 PM	475	0
6:00 PM	500	0
8:00 PM	600	0

This situation could lead to rejection of new client connections in near future. This shows that the load balancer is not working as desired. The entire concept of RAC may fail when load balancing doesn't work. The present research work is a remedial solution for getting alert message beforehand and temporarily managing such situation by following manual switchover.

Although this is not a permanent solution of the problem but it could be very helpful for mission critical databases, where rejection of connection or slow retrieval of data could harm the business.

5. CONCLUSION

In this study, the performance of load balance monitoring of Oracle RAC using relative entropy calculation is evaluated. The Relative Entropy calculation scheme of detecting the randomness of Oracle RAC system is a truly new approach based on which we can monitor as well as forecast the behavior of its load balancing act. Proposed method also could help to detect any upcoming issue related to load balancing which could interrupt the client services. This research work is performed on a two node RAC database, this can be extended more to incorporate in finding relative entropy based monitoring of load balance in multiple node Oracle RAC database.

6. ACKNOWLEDGMENTS

Thanks to our family and friends particularly Mr. Bijesh Balagopal and Mr. Narayana Kumar S. who directly or indirectly contributed in the research work.

7. REFERENCES

[1] docs.oracle.com/cd/E11882_01/rac.112/e41960/admcon.htm

[2] Gong Weihua and Wang Yuanzhen, Jan, 2006 "A new load balancing scheme on heterogeneous database cluster" Geo-spatial information science; 9(3):216-222. DOI:10.1007/BF02826771

[3] Stephane Ganc, Arski and Hubert Naacke "Load Balancing of Autonomous Applications and Databases in a Cluster System" Distributed data and structure 4, 162-164 -169.

[4] Zin Chen, Gokul Soundrajan,Madlin Mihaailescu 2007, "Outlier Detection for Fine-grained Load Balancing in Database Clusters" Proceeding ICDEW 07, Proceedings of 2007 IEEE International Conference on Data Engineering workshop, IEEE Computer Society Washington, DC, USA, 404-413

[5] Don MacVittie "Load Balancing Oracle Database Traffic" F5 Networks, Inc.White paper, 8-10, 888-882-4447

[6] Gia- Khanh Nguyen and Tim Read, July 2011 "Running oracle® real application clusters on oracle solaris zone clusters" An Oracle White Paper

[7] Deepali Kadam, Nandan Bhalwarkar, Rahul Neware, Rajesh Sapkale, Raunika Lamge, June-2011, "Oracle Real Application Clusters" Int. J of Sc. & Engg Research , 2, (6), 01-05, ISSN 2229-5518

[8] Rene Kundersma, 2012, "11gR2 Oracle Real Application Clusters Grid Infrastructure N.F." otn.oracle.com/rac

[9] Howard Karlof and Kenneth Shirley, 2013, "Maximum Entropy Summary Trees" Eurographics Conference on Visualization (EuroVis) Vol 32, No. 3

[10] Markus Michalewic, June 2013, "Oracle Database 12c Real Application Clusters (RAC)" An Oracle White Paper,15-17