Performance tuning:

I work in trading domain. My Client name is BNP, my Project / application name is **sharekhan**. My customers are **retail & institutional clients**. Retail clients trade by their own money and institutional clients trade by their client's money. In sharekhan page, client can placeonly one order at a time. While place the new order, **sp_trade_now** procedure will be called by the java developers(in **API**) and the inputs will be stored in the **t_orders** table.The inputs of the table or column of the table are **sl.no, exchange, script, qty,Market price,Limit price**.Mostly retail clients use this page to place their order.

Now our new **work request** is,

The institutional clients ask for a new page to place their bulk orders. So, client asked us to prepare the new page called **bulk orders**. We (pl/sql developers) usually get the WR **monthly once**. we understand the BRD given by BA and table design document prepared by the data modeler. And bulk orders front pagecreated by the Java developers. We asked java developers to call the previous procedure itself. While placing the bulk orders, for example 1000 inputs, the sp_trade_now procedure will loop for 1000 times and 1000 rows will be inserted in t_orders table. So, performance got slow.

L1 team interact with the customers and raised **the incident request (bugs in WR).**The incident request is, the query is running for 4 hours i.e. customers got the order placed successfully feedback after 4 hours only. L1 team set the priority level and they escalated t L2 team.

L2 team analyzed the problem and found this is because of **table lock**.Whenever more than one user accesses the table in same time, table will be locked.To avoid the table lock,user must commit or session to be killed. L2 team found the locked object using **v$locked_object** and **all_objects.**

Query to find the locked object is,

Select b. owner, b.object_name, a.oracle_username, a.os_user_name from v$locked_object a, all_objects b where a.object_id=b.object_id;

So L2 team escalated the **IR** toL3 team, to kill the session.L3 team get the session_id(sid) and serial number of that session from **v$session**and killed the session using the following the command ang gave the temporary fix.

Alter system kill session'sid,serial#';

Problem request(PR):

The repeated IR's will be raised as a PR by the L1 team for the permanent fix.Again, L2 team analyzed and asked the L3 team to create the new procedure called **sp_bulk_orders** and new table called **t_xml_orders having only one column named as orders in xmltype datatype** and asked the java developers to **call this new procedure**(**in API**)and give the **input in xml**.

Now the procedure will be called only one time and it will store only one row in a xml rather than 1000 rows. Now the query is running for **1 second only**. Like this we did performance tuning to reduce the running time of the query.

Java developers will **generate the xml** from the customer's 1000 orders input and give us as an input. We will get that xml as a string using **clob datatype and store in t_xml_orders as a xml using xmltype datatype.** Then we will give the output to java developer's in **refcursor using sys_refcursor** datatype by extracting the xmltype using **extract function and xmlsequence function.**Refcursor is a datatype to store the result of the query.mostly refcursor Is used to return the results to the client.

Last WR : bulk orders

Last IR and PR : query is running for 4 hours to produce the order placed successfully output.

Last created stored procedure : sp_bulk_orders

Last created table : t_xml_orders

Last used datatypes in that procedure and table : xmltype,refcursor,clob