ABSTRACT

Web application usages are wider since they can access over the network. Attaining reliable performance in the web application is primary challenging factor. Web application performance suffers due to many reasons. Based on this research and analysis, Network traffic congestion and heavy load on database server are found to be the primary reasons and they cause the web application performance degradation.

In the first case, Web application users are facing this performance issues while accessing the web applications via URL which were hosted on a web server. Traversal from the user-end may take numerous gateways, hubs and proxies in order to access the web server. Many times, unnecessary routing to proxy servers creates degradation in the performance while accessing the web server. Also, the collision and contention lead to poor performance in accessing the web server by many users through the proxy servers. This also induces an additional latency in accessing the destination URL.

In the second case, in three tier (App, Web and Database) architecture, Web application performance mostly suffers in getting the reliable response from the database server. Web application performance degrades severely due to the surplus congestion on database server. Congestion on database server is made due to more OLTP and OLAP related
tasks. Also, continuing growth of data volume increase and number of concurrent users accessing those data increases performance related issues. This database server bottleneck may sometimes lead to slow down or to hang the entire web applications.

Increasing hardware resources (IO, Memory and CPU) are not enough to acquire good performance in web applications unless and until application Queries and Stored Procedures are optimized towards the server performance. Poorly designed query may harm the entire server performance and eventually lead to affect or even to stop entire business. Even well-designed applications also may experience performance problems if the SQL is poorly tuned. Such slow performance in the database leads the web applications to perform very slowly. As a result, web server response time also seems to be very slow.

Web applications use TempDB as the global resource for the SQL server system. TempDB is one of the primary system databases that extremely decide the reasonable performance of the SQL server instance and the server. Impact, in terms of heavy load on the TempDB, may lead to the performance issue in entire system due to the resource crunch.

Two regions have been identified in the web application environment to deal with all the above real time performance issues and to improve the web application performance. First one is traffic bypassing in the network and the next is reducing the heavy load on the database server in order to improve the web application performance. For achieving this, there
may be a need to tweak few configuration settings on network and database server regions.

In view point of the network, in the present work; traffic bypassing is implemented using iRules on a F5 load balancer. Through this methodology, it is possible to configure the F5-load balancer to bypass the proxy servers and can directly access to the destination server. Also, additional latency could be avoided by certain trusted users/network. Furthermore, to deal with the security vulnerability issues, in the present work, NS-IDS security system is implemented in the web server to restrict the access of the network by anonymous users.

On the database server region, the methodology used in the present work is to separate OLAP, reports and workloads into standby server and OLTP workloads on primary database server not to impede each other. Ultimately, web application performance will be improved by getting reliable response from database server by balancing the load on the production database servers. Based on this methodology, OLTP and reports related request from web server shall be processed in effectively by database server. At the same time, the standby server resources (CPU, IO and RAM) also will be utilized effectively.

Database server performance improvement can be achieved by reducing the load on TempDB in the SQL server. Query and Stored Procedures optimization is done in order to reduce the TempDB load and to improve the overall production database server performance. The ultimate
aim is to improve the web application performance in terms of high response
time of web server by improving the database server performance.

By reducing the traffic in the network and the load on the database
server, the web server response time can be improved efficiently. Ultimately,
web application is improved.