CHAPTER 3

PROBLEM STATEMENT AND RESEARCH METHODOLOGY

Web application operations are a crucial aspect of most organizational operations. Among them business continuity is one of the main concerns. Companies rely on their information systems to run their operations. If a system becomes unavailable, company operations may become handicapped or stopped completely. It is necessary to provide a reliable infrastructure for information technology oriented operations, in order to minimize any chance of disruption or disaster and to sustain the adequate performance of the systems. Major challenges are to have the data availability at all times with reliable network traffic in the web architecture and heavy load on the database server are leading the performance hindrances of web application.

3.1 PROBLEM STATEMENT

Web application performance suffers due to many reasons in which, below given are the primary reasons that cause the web application performance degradation. The Figure 3.1 shows the factors of web application performance degradation.

- Network Traffic Congestion
- Heavy Load On Database Server
In these days, web application usages are wider since they can access over the network. One primary hurdle when accessing the web applications over the network is higher latencies due to poor performance. Users are facing this performance issues when accessing the web applications via URL which were hosted on a web server. Traversal from the user-end can take numerous gateways, hubs and proxies in order to access the web server. Client/users IPs are facing performance hindrances while accessing the web server, due to unnecessary routing to proxy servers from the F5-load balancer. All the users are intended to route to the proxy servers where they will face the collision and contention that lead to poor performance to access the web server.
Higher Latencies in accessing the web server

- When users start accessing the application, the entire traffic coming from user WAN link goes to proxy server where the content scanning takes place and then the traffic comes back to load balancer then it moves to the actual destination server through a firewall.

- The major issue with proxy server is that they will face the collision and contention that lead to poor performance when accessing the web server and which is inducing an additional latency in accessing the destination URL.

F5-Load Balancer

F5-load balancer is a device that acts as a reverse proxy and distributes network or application traffic across a number of servers. Load balancers are used to increase the capacity (concurrent users) and reliability of applications. They improve the performance of applications by decreasing the burden on servers associated with managing and maintaining application and network sessions, as well as by performing application-specific tasks.

Load balancers are generally grouped into two categories: Layer 4 and Layer 7. Layer 4 load balancers act upon data found in network and transport layer protocols (IP, TCP, FTP, UDP). Layer 7 load balancers distribute requests based upon data found in application layer protocols such as HTTP.
Requests are received by both types of load balancers and they are distributed to a particular server based on a configured algorithm. Some industry standard algorithms are:

- Round robin
- Weighted round robin
- Least connections
- Least response time

Round Robin passes each new connection request to the next server in line, eventually distributing connections evenly across the array of machines being load balanced. Round Robin works well in most configurations, but could be better if the equipment that you are load balancing is not roughly equal in processing speed, connection speed, and/or memory.

Layer 7 load balancers can further distribute requests based on application specific data such as HTTP headers, cookies, or data within the application message itself, such as the value of a specific parameter.

Load balancers ensure reliability and availability by monitoring the "health" of web applications and sending requests to servers and applications that can respond in a timely manner.

BIG-IP network appliance was originally a network load balancer. F5's BIG-IP product is based on a network appliance (either virtual or physical) that runs F5's Traffic Management Operating System (TMOS), which runs on top of Linux.
Figure 3.2 Architecture of web server access through the F5-load balancer

Architecture of F5 load balancer

Web application hosted and access architecture was illustrated above in the Figure 3.2.

1. The traffic from customer WAN link is entering the F5 load balancer through the interface 00

2. And the customer traffic was moving to proxy servers 01 and 02 through Load balancer interface 02 and via switch 01.
3. Once the content scanning is done by proxy server, the traffic was coming back to Load balancer via the interface 03 through switch 02.

4. Then scanned traffic received by Load balancer from proxy servers was being sent to the Web server via the interface 01 through the firewall.

The major issue with proxy servers is they will face the collision and contention that lead to poor performance when accessing the web server and which is inducing an additional latency in accessing the destination URL.

3.1.2 Heavy Load on Database Server

Heavy load on database server is also degrading the web application performance. Database server load can be separated as per the following natures.

- Load due to OLTP and OLAP tasks
- Load due to querying tasks.

3.1.2.1 Load due to OLTP and OLAP Tasks

Performance hindrances on the web applications are major challenges faced by IT solution providers. In three tier (App, Web and Database) architecture, web application performance mostly suffers to get the reliable response from the database server. Web application performance behaves poorly due to the surplus congestion on database server. Congestion on database server due to more OLTP and OLAP related tasks that lead to poor performance of web applications.
Disaster Recovery Mechanism in SQL Server

Importance of disaster recovery planning is, as IT systems have become increasingly critical to the smooth operation of a company, and arguably the economy as a whole, the importance of ensuring the continued operation of those systems, or the rapid recovery of the systems, has increased.

It is estimated that most large companies spend between 2% and 4% of their IT budget on disaster recovery planning, with an aim of avoiding larger losses in the event that the business cannot continue due to loss of IT infrastructure and data.

Log shipping is one of the database disaster recovery techniques which are available in the SQL server. It is used to synchronize the distributed database server and which can improve the application performance and availability of database. Synchronize the database by copying transaction logs, backing up, and restoring data. SQL server used SQL server job agents for making those processes automatic. Log shipping does not involve automatic transfer of server if there is any failure. If the primary server fails, it will not redirect the application to a standby server. This has to be done manually.
Heavy Load on Primary Database Server Description

Web application users are accessing primary database server through requesting web server. And Online Transactional Processing (OLTP), Online Analytical Processing (OLAP) and Reports related transactions are processing at primary database Server. Ultimately, primary database server resources (Memory, IO and Processor) are utilized heavily.

All the SQL procedures which related to the OLTP, OLAP and reports execution happens on the primary database server. Due to this, database and tables of primary server were heavily accessed.

Standby server resources are not efficiently utilized apart from the operation of database synchronization.

Primary database server contains mixing of OLTP, OLAP, and reporting workloads hugely. OLTP workloads are characterized by many small transactions, with an expectation of quick response time from the user. OLAP and reporting workloads are characterized by a few long-running operations that might consume more resources and cause more contention. The long-running operations are caused by locking and by the underlying physical sub-system.

Hence, the long running operations cause the performance hindrances of web application. Due to this response from database server to web server takes plenty amount of time (Figure 3.3).
3.1.2.2 Load due to Querying Tasks

Performance problem arises in database server when continuous growth of data volume and number of concurrent users to access those data. Database server bottleneck may lead to stoppage of entire web applications. Increasing hardware resources (IO, Memory and CPU) is not enough to acquire good performance in web application unless and until applications Queries and Stored Procedures are optimized towards the server performance. Poorly designed query may harm the entire server performance and eventually lead to entire business stoppage. A well-designed application may still experience performance problems if the SQL it uses is poor tuned. Due to
slow performance in the database, web application also performs very slowly and web server response time also seems to be very slow.

**TempDB Database in SQL Server**

The TempDB system database is very similar to a user database. The main difference is that data in TempDB does not persist after SQL server shuts down. SQL server uses TempDB to store internal objects such as the intermediate results of a query. It is a workspace for holding temporary objects or intermediate result sets.

All the production user databases in the SQL server instance will access the same TempDB database only. Hence, if any problematic query from any one user database may acquire more resources (IO, Memory, and CPU) on TempDB which leads to poor performance of other databases, applications and entire system in the same SQL instance. Therefore, one needs to be more cautious on designing the user queries on the user databases with respect to the TempDB performance.

In the SQL server, query execution and processing the query will acquire the CPU, memory and disk resources.

**How TempDB Database is affecting the Server Performance**

- Since it is a global resource of the SQL server system, the entire user created databases and users can have access to this TempDB database.

- Log and data files growth on TempDB may lead to the server running out of space.
• Responsible for page and extent allocations and DE allocations for all type of objects in TempDB and this may lead to the DML operations contention.

• Responsible for manipulation of metadata when user objects in TempDB are created or dropped and this may lead to DDL operations contentions.

• DDL and DML operations contention may acquire high utilization of CPU and memory may lead to SQL server services restart and cause overall system stoppage.

• There are many production user databases in the SQL Server instance will access the same TempDB database only. Hence, if any problematic query from any one user database may acquires more resources (IO, Memory, and CPU) on TempDB which lead poor performance of other databases, applications and entire system in the same SQL instance.

**Primary Operations that Leads to Heavy Load on TempDB**

• Heavy usage of temporary tables (Local and Global).

• Heavy usage of temporary variables.

• Heavy usage of CTE (Common table Expressions).

All the above options are used for reports related procedures and SQLs as they can be used in multi session and they are used most often to provide workspace for the intermediate results when processing data within a batch or procedure. During peak hours, this workload on TempDB may affect the entire OLTP transactions of the system.
TempDB is the global resource for the SQL server system. It 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 leads to the performance hindrances of entire system due to the resource crunch.

3.2 RESEARCH OBJECTIVES

The objectives of this research are defined in such a way to achieve the goal of the research and they are stated as follows:

1. Improving the web application performance by bypassing the traffic on F5-Load balancer.
2. An effective data transmission in Internet using NS-IDS to improve the performance.
3. Lessening the load on data base server
4. Accelerating the data transmission with security mechanism.

3.3 RESEARCH METHODOLOGIES

The new research methodologies are used to achieve the objectives in the current work. These methodologies are represented in the following architecture (Figure 3.4).
3.3.1 Improve the Web Application Performance by Bypassing the Traffic on F5-Load Balancer

Traffic bypassing using iRules on a F5 load balancer can reduce the network traffic. Using this methodology configured the F5-load balancer to bypass the proxy servers and can directly access to the destination server. Through this, additional latency is avoided for certain trusted users/network.

3.3.2 An Effective Data Transmission in Internet using NS-IDS to Improve the Performance of Web Application

After implementing traffic bypassing using F5-load balancer, the scholar may come across the situations of security violation in the network
and anonymous users can access the web server. For this reason, the research scholar implemented the NS-IDS security system in the web server to restrict the access of anonymous users from the network.

- NS-IDS limit the spread of Trojans and from the network.
- NS-IDS prevent communication outside business area and limit the scope of intrusion of unauthorized.
- Information leaks are also reduced drastically as NS-IDS control all the traffic in the Network.

An Intrusion Detection System (IDS) is a software component which needs to be installed on web server or any network components. This monitors the network traffic and logs their information in periodically. The IDS takes action against the anonymous and malicious users in the network. There are varieties of flavors available with NS-IDS to accomplish the security related tasks.

3.3.3 Lessening the Load on Data Base Server

This methodology applies the following two steps for performance progress in the database server.

1. Performance improvement of web applications in disaster recovery mechanism to separate the OLAP and OLTP loads.

2. Reducing SQL server TempDB load by optimizing queries and stored procedures in user databases for improving overall production database server performance.
Performance improvement of web applications in disaster recovery mechanism to separate OLAP and OLTP loads

Main intention of this methodology is to separate OLAP, Reports workloads into standby server and OLTP workloads on primary database server not to impede each other. Ultimately, web application performance is improved by getting reliable response from database server by balancing the load on production database servers.

Reducing SQL Server TempDB Load by optimizing Queries and Stored Procedure in user Databases for Improving Performance

This methodology optimized the Query and Stored Procedures in order to reduce the TempDB load and 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.

3.3.4 Accelerating the Data Transmission with Security Mechanism

The best possible data transmission performance with security, successfully established through this methodology when the network between the sender and the receiver is maintained full of data, i.e. ‘in flight bytes’ is set to the maximum that a sender and receiver shall be in buffer period and an acknowledgement is issued.
3.5 SUMMARY

The performance of the Web Server/applications access will be improved by configuring iRules on F5-load balancer to bypass the Proxy servers and can directly access to the destination server. In the database server aspects, reliable performance will be gained by separating OLAP, Reports workloads into standby server and OLTP workloads on primary database server not to impede each other. The web application performance with ensured network reliability and security has been improved through the entire proposed methodologies.