



Web applications Performance Symptoms and Bottlenecks Identification

Abstract

Performance bottleneck identification is an uphill task in web applications testing because there could be various reasons behind a slow performing web application. Performance bottlenecks identification requires complete non-functional (architectural and behavioral) information of the application as well as necessary input from all stakeholders and exhaustive performance testing activity to achieve the desired outcomes. In this paper we will identify various performance symptoms, potential problematic areas and bottlenecks of each web application component which needs to be focused during performance testing.



Introduction

Application current state against its performance goals is found out by conducting performance testing and its behavior in expected and unexpected conditions in production environment is also measured by the same testing process. Performance testing ensures that all the application bottlenecks are identified which will cause bad user experience. It is hard to identify performance bottlenecks without prior knowledge of all the possible performance issues and their symptoms in each problematic area of the application. Unless all the performance bottlenecks and their root causes are identified, it's tough to make the application perform better.

Performance testing is a team effort where all stakeholders like business owner, business analyst, marketing team, network team, development team, QA and performance testing team participate and involve making it happen. Performance tester needs stakeholders to give proper input to understand the application architecture, its user's behavior and application performance goals in order to successfully isolate the required load and to identify performance bottlenecks successfully.

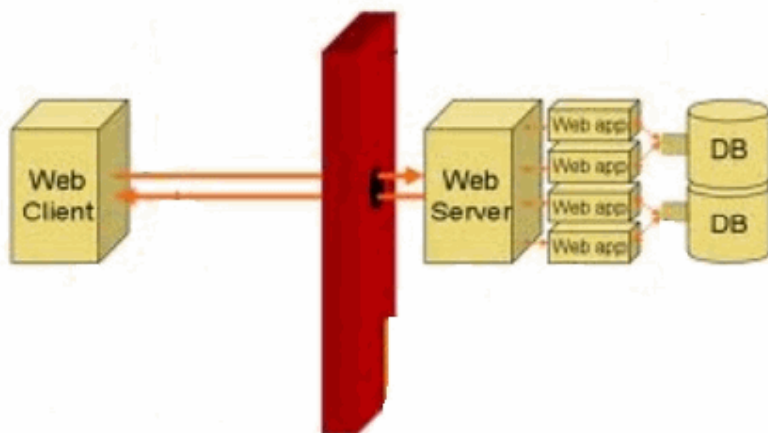
A performance testing report that simply state "Application is not performing to the expectations" is not enough. You have to provide the reasons to justify its weak performance and solutions to improve the same for desired results. Stakeholders will be more interested in knowing "Why application is slow" and "What should be done to fix the issues" rather than just knowing the basic information only. Therefore, identifying the bottlenecks and their root causes is the core of performance testing and a successful activity will always provide complete details of these questions.

Following we will identify sources of bottlenecks, typical performance symptoms and major bottlenecks of every component of web application architecture.

Web Applications Architecture

It is important to understand the web applications architecture before listing their performance bottlenecks because it will help in understanding the impact of every bottleneck.

In software engineering, different client-server architectures are used for web application development. These architectures logically differentiate among data presentation, application processing and data management functions. 3-tier architecture (web server, application server and database server) is the most famous N- tier client-server architecture used for web application development at enterprise level.



Following is the brief description of each components of the 3-tier web application architecture.

Network Devices

Although network devices don't make any tier of the 3-tier architecture but different network devices like network interface card, firewall, cables, load balancer, routers etc. are used to support and connect other components.

Web Server

Web server which is on 1st tier in 3-tier architecture consists of low capacity computer(s) that receive user requests, send them to required server and show received results to users.

Application Server

Application server is 2nd tier in 3-tier architecture having one or more medium capacity computer(s) that receive user requests from the web server, apply business logic on them and send them back to the web server.

Database Server

Database server is in the last tier of 3-tier client-server architecture which normally consists of a high capacity computer with stand-by facility that manages database access to facilitate user data requests.

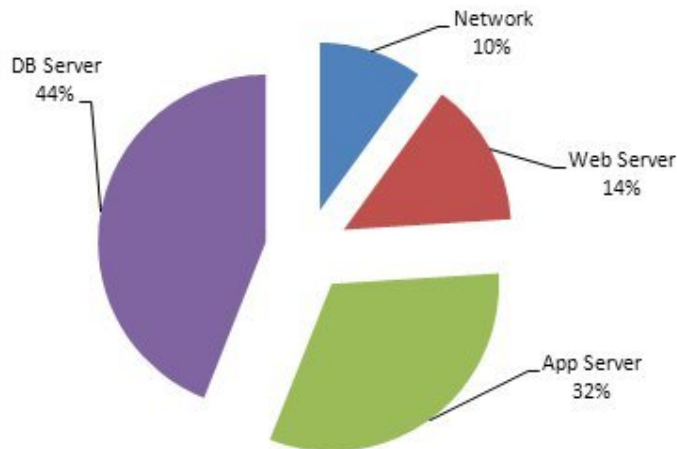
Sources of Bottlenecks

As discussed the web applications architectural components (web server, application server, database server and network resources) in our previous section, these are the potential areas where most of performance bottleneck resides.

It is tough to test each and every component's performance thoroughly. Server hardware and network resources are assumed to be the main culprits for lower performance. Especially, server upgrades are considered as the best source of performance optimization and that's why this old saying "When all else fails, throw more hardware at it" is recalled. However, different studies and experiences reveal



that it is the application code which mainly causes the performance bottlenecks. Based on experience, following are the statistics of performance bottlenecks probability in each of its component.



These results show that 76% of the performance bottlenecks appear in the application and database servers where most of the application code resides and this code causes most of the issues. Meanwhile, less percentage of bottlenecks is produced due to web server and network resources. Identifying bottleneck source is important to optimize the application performance'.

Web Applications Symptoms of Performance Bottlenecks

Once all the performance bottleneck sources have been spotted, next step is to identify the application performance issues. People assume that conventional performance optimization techniques (i.e. increase server memory, optimize application code, displace the server, change the database indexing, network/internet up gradation etc.) without identifying the root cause, will be sufficient enough for improvements. These techniques may work sometimes but do not guarantee a permanent fix and mostly they end up with wasting a lot of efforts and money.

Diagnosing the web application performance problems is a demanding task. An application would have various types of issues like functional, usability, security, cross browser compatibility and performance etc. In such situations, it is extremely important to differentiate the performance issues from others appropriately.

There is a long list of web applications potential performance bottlenecks and some of them are as following,

- Extended response time of user
- Extended response time of server
- High CPU usage
- Invalid data returned
- HTTP errors (4xx, 5xx)
- Lots of open connections



- Lengthy queues of requests
- Memory leaks
- Extensive table scans of database
- Database deadlocks
- Pages unavailable

Web Applications Performance Bottlenecks

Application architecture is formed by several components and there could be dozens of bad performance symptoms in each component. Being a good performance tester, one must know the list of performance symptoms on each tier to diagnose bottlenecks effectively.

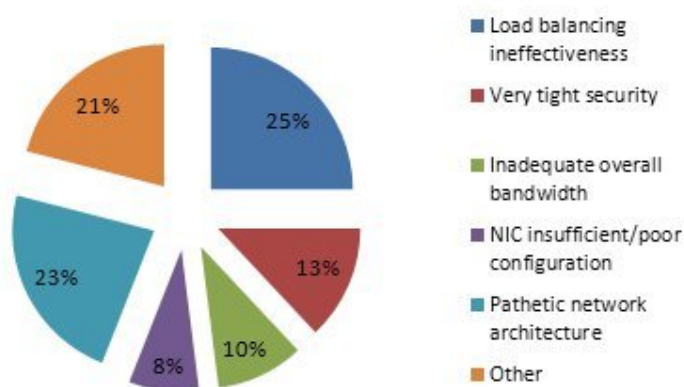
Below is the detailed list of symptoms of each of the web applications 3-tier component.

Network Performance Bottlenecks

Network bottlenecks contribute very little however they are important enough to be discussed in detail because you cannot afford minor issues as well because they can lead to disasters. Following are the major network performance symptoms in context of 3-tier web applications,

- Load balancing ineffectiveness
- Network interface card insufficient/poor configuration
- Very tight security
- Inadequate overall bandwidth
- Pathetic network architecture

Network performance bottlenecks don't have any certain source. Load balancing, security and network architecture can be the major sources. Below pie chart depict percentage of each source to illustrate their impact on performance bottlenecks.



Web Server Performance Bottlenecks

Like network performance bottlenecks, web server bottlenecks don't have major contribution to the performance issues as well. Web servers act as a liaison between client and processing servers

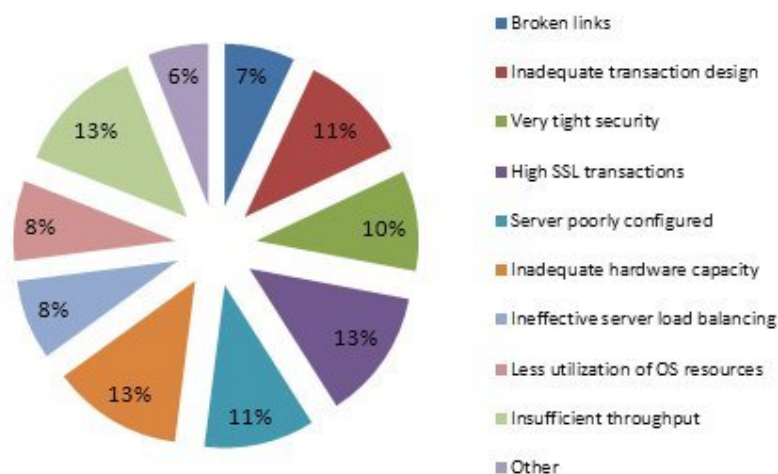


(application and database). So web server performance bottlenecks need to be addressed properly since they can affect other components performance to great extent.

Below is the list of bottlenecks which can affect web server performance,

- Broken links
- Inadequate transaction design
- Very tight security
- Inadequate hardware capacity
- High SSL transactions
- Server poorly configured
- Servers with ineffective load balancing
- Less utilization of OS resources
- Insufficient throughput

Secure transactions has major contribution to web server performance bottlenecks however usually it is load balancing as well and sometimes high resource specialized functions cause web server performance. Below is the graphical representation of each web server performance bottlenecks with percentage.



Application Server Performance Bottlenecks

Business logic of an application resides on the application server. Application server hardware, software and application design can affect the performance to great extent. Poor application server performance can be a critical source of performance bottlenecks.

Below is the list of application server bad performance causes,

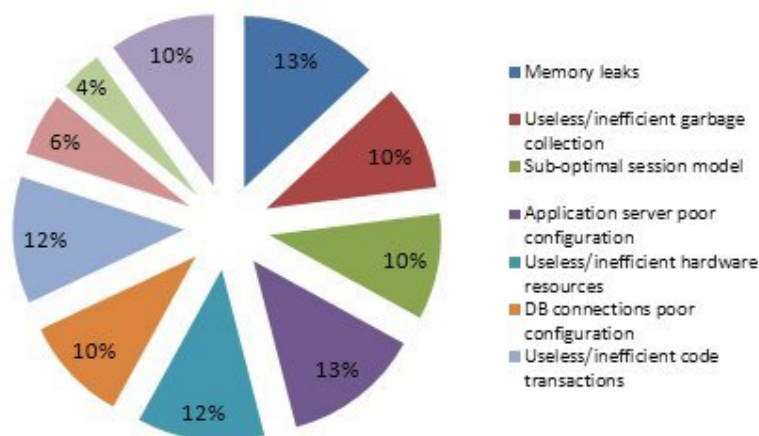
- Memory leaks
- Useless/inefficient garbage collection
- DB connections poor configuration
- Useless/inefficient code transactions
- Sub-optimal session model



- Application server poor configuration
- Useless/inefficient hardware resources
- Useless/inefficient object access model
- Useless/inefficient security model
- Less utilization of OS resources

Object caching, SQL and database connection polling are the main causes of application server bottlenecks and they contribute 60% to the application server. 20% of the times inefficient application server causes poor performances.

Below is the complete detail of application server bottlenecks with their impact.



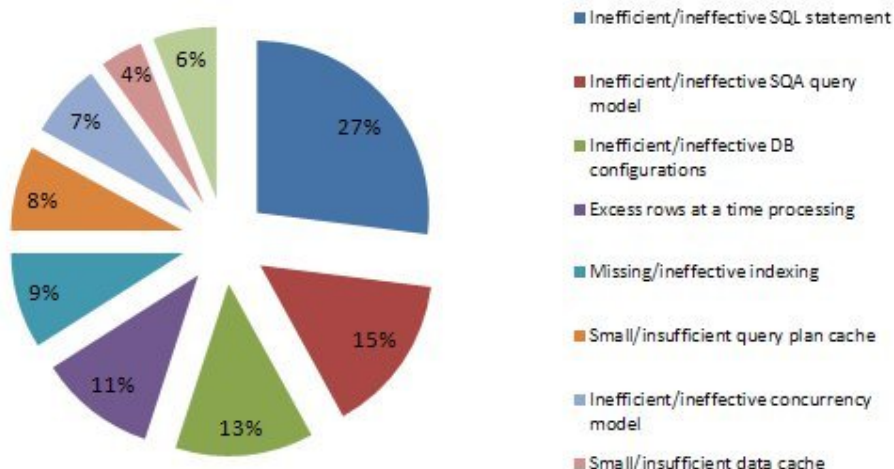
Database Server Performance Bottlenecks

Database performance is most critical for application performance as this is the main culprit in performance bottlenecks. Database software, hardware and design can really impact the whole system performance.

Following is the comprehensive list of database poor performance causes,

- Inefficient/ineffective SQL statement
- Small/insufficient query plan cache
- Inefficient/ineffective SQA query model
- Inefficient/ineffective DB configurations
- Small/insufficient data cache
- Excess DB connections
- Excess rows at a time processing
- Missing/ineffective indexing
- Inefficient/ineffective concurrency model
- Outdated statistics
- Deadlocks

Bad SQL and indexes contributes nearly 60% to the database server performance bottlenecks. Below chart will show complete detail of database server causes with percentage.



Client Side Performance Bottlenecks

The client side performance aspects as undercome an increased interest in the last years with the release of Google performance optimization best practices like caching, lesser number of static files, file minification, compression, java script processing time, page rendering, etc.

For rich internet applications with lots of images, videos, etc, the client side aspects have a bigger bearing on the actual response time when compared to the server side response time and should be given due importance.

Using many modern AJAX architectures, it is possible to place so much code in the client that a significant amount of time is required before the request is transmitted to the application server. This is particularly true for underpowered client machines with inadequate memory and slow processors.

Top 10 client side performance symptoms:

- Slow CSS Selectors on Internet Explorer
- Slow executing external services
- Multiple CSS Lookups for same object
- Extensive XHR Calls
- Large DOM
- Expensive DOM Manipulations
- Extensive Visual Effects
- Extensive JavaScript files
- Extensive Event Handler Bindings
- Too fine granular logging and monitoring

Third Party Services Performance Issues



Today web applications heavily rely on third party components which affect the page loading and result in bad user experience. It is a common practice that third party tools are not properly analyzed from performance point of view before integrating them into the application. If you ever observed the page components load time, you would have experienced that third party components take more time. These third party components can cause various performance bottlenecks but following are the most common,

- Size of page increases
- Third party services utilize more bandwidth utilization
- Not decreased resources
- Inadequate response time of third party component provider

Conclusion

Web applications are becoming complex day by day and at the same time, the identification of their performance bottlenecks is becoming a tough task. Various factors contribute to the performance issues and knowledge of their issues & their symptoms is mandatory to rectify the performance bottlenecks. Web server, application server, database server hardware, software and design along with network configurations can be major contributors to performance bottlenecks. Moreover, client side design and third party components can also affect web application performance. Knowledge of a complete list of symptoms of all potential problematic areas will help in identifying the root cause and its remedy.

Sajid Manzoor