

Abstract

In this paper we present a survey on web servers – IIS, Apache, Sun Java web server, Apache Tomcat. Our survey work involves a comparative study of these web servers with respect to the following parameter: Performance, Scalability, Web server management, Dynamic content support, Security.. At the end, a study of web servers has been made by comparing the above mentioned web servers against all the parameters mentioned.

1. Introduction

A Web server is a computer program that is responsible for accepting HTTP requests from clients (user agents such as web browsers), and serving them HTTP responses along with optional data contents, which usually are web pages such as HTML documents and linked objects (images, etc).

There are many web servers in today's market namely Apache, Websphere, Internet Information Server, IPlanet, Tomcat, nginx, GWS. We chose only 4 web servers for comparison which are easily available. They are IIS, Apache, IPlanet and Tomcat.

We started with literature survey on all the four web servers. After a thorough study about the web servers, we categorized some of the parameters for comparison of the web servers. We need some tools for the comparison of the web servers based on the categorized parameters. Then, we did study on different tools available for comparison of web servers. We chose JMeter tool for comparison of various parameters among the web servers. We have compared each parameter for all the four web servers. The conclusion for each parameter is also mentioned in our report.

2. Evaluation Method: Why are we different?

Our evaluation method went through the following phases.

Phase 1: Literature survey on web servers

Phase 2: Selection of web servers for comparison

Phase 3: Finalization of parameters for comparison

Phase 4: Finalization of tools for evaluation of parameters

Phase 5: Evaluation of parameters, obtain results

Phase 6: Conclusion for each parameter

We started with literature survey on various web servers. Market share of various web servers were studied. We chose four web servers; IIS, Apache, Sun one Java Web server, Tomcat depending on the market share and the availability. In the next phase, parameters were finalized for comparison, followed by selection of tools for evaluating parameters. Finally, each parameter was evaluated and results were obtained.

3. Features of Web Servers

Some of the important features of Web Servers are

- **HTTP:** every web server program operates by accepting HTTP requests from the client, and providing an HTTP response to the client. The HTTP response usually consists of an HTML document, but can also be a raw file, an image, or some other type of document (defined by MIME-types).
- **Logging:** usually web servers have also the capability of logging some detailed information, about client requests and server responses, to log files.
- **Authentication,** optional authorization request (request of user name and password) before allowing access to some or all kind of resources, HTTPS support.
 - **Basic authentication:** Basic Authentication is the same as the http process of authentication. All transactions are in clear text, but usernames and passwords are encoded.
 - **Digest authentication:** This method of authentication is safer than basic authentication as the user credentials are hashed or encrypted. Authentication that is processed in the Digest manner involves the user credentials passing through a one-way process, also known as hashing.

- SSL authentication support.
 - HTTP Filtering: IP based filtering.
 - URL Authorization: to manage access control for Web-based or line-of-business applications in enterprise environments.
-
- Handling of **static content** (file content recorded in server's filesystem(s)) and dynamic content by supporting one or more related interfaces (SSI, CGI, SCGI, FastCGI, JSP, PHP, ASP, ASP.NET, Server API such as NSAPI, ISAPI, etc.
 - **Content compression** and large file support to be able to serve files whose size is greater than 2 GB on 32 bit OS.
 - **Virtual hosting** is a method that servers such as web servers use to host more than one domain name on the same computer, sometimes on the same IP address.
 - **Large file support** to be able to serve files whose size is greater than 2 GB on 32 bit OS.
 - **Bandwidth throttling** to limit the speed of responses in order to not saturate the network and to be able to serve more clients.

4. Parameters for Web Servers Benchmarking

The following parameters were chosen for Web Servers benchmarking

- Performance
- IP Blocking
- Self Signed SSL certificates
- Handling new file extensions
- Hiding file extensions (URL rewriting)

5. Webservice Benchmarking Tools

We came across many web server benchmarking tools; such as Httpperf, autobench, OpenSTA, ApacheJmeter, ab (apache bench) , grinder performance testing tool, webload , openwebload , netwox, WebServer Stress tool. We have used autobench and ApacheJMeter.

5.1 Benchmark System

We tested web servers for the above mentioned parameters on Windows Server 2003.

6. Results

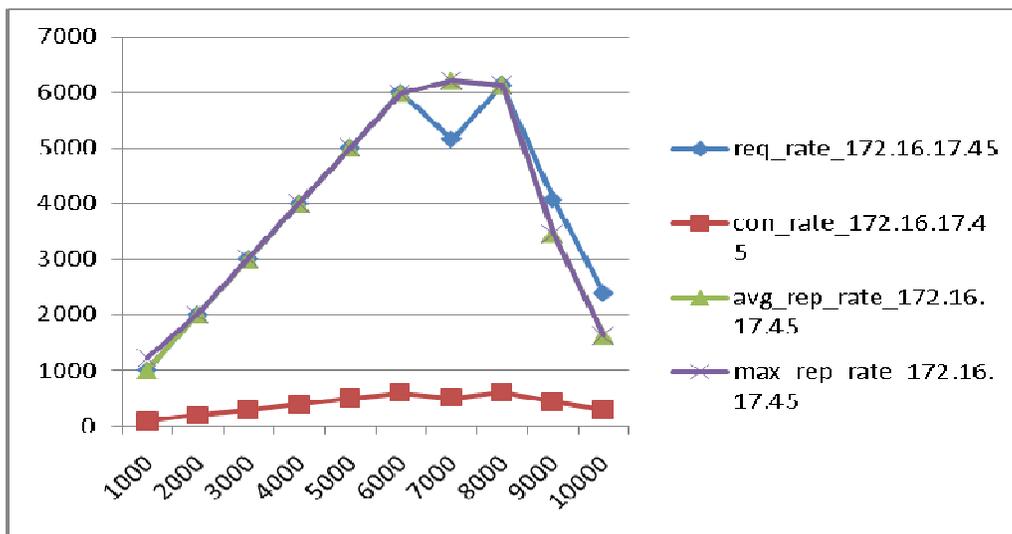
6.1 Performance (Load Testing)

We carried out many tests on IIS, Apache, Apache Tomcat and Sun Java web server. We tested for static pages and dynamic pages. The results obtained are as shown below.

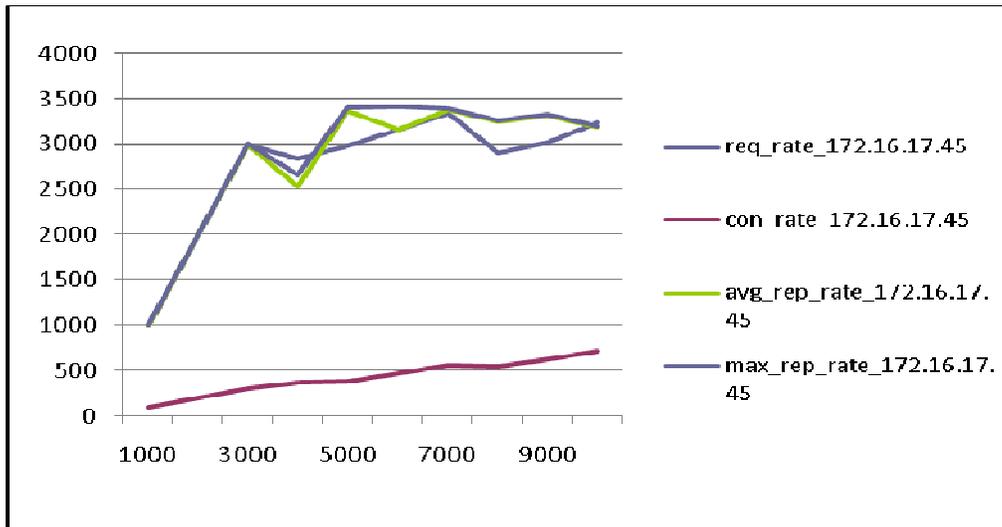
Here values on X axis for all the graphs is demand request rate.

Static Pages

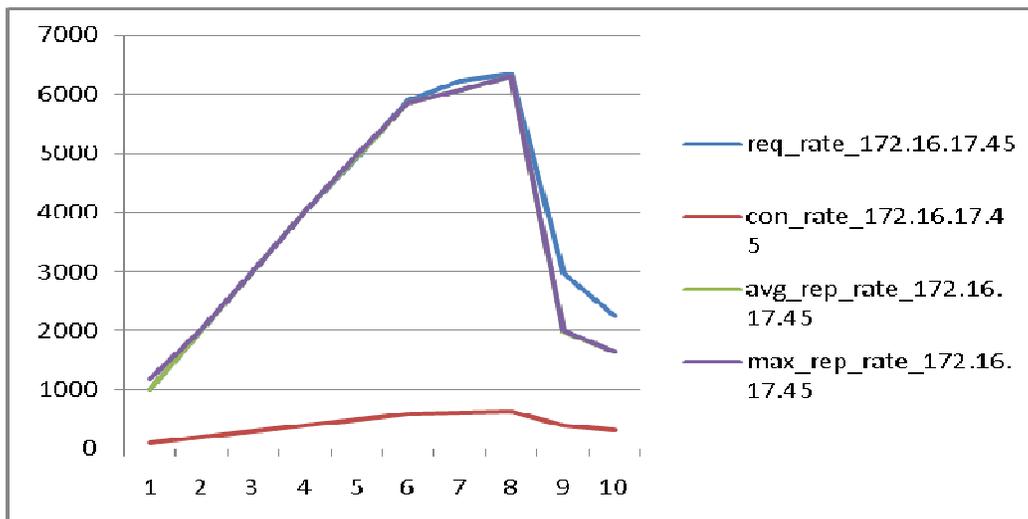
IIS



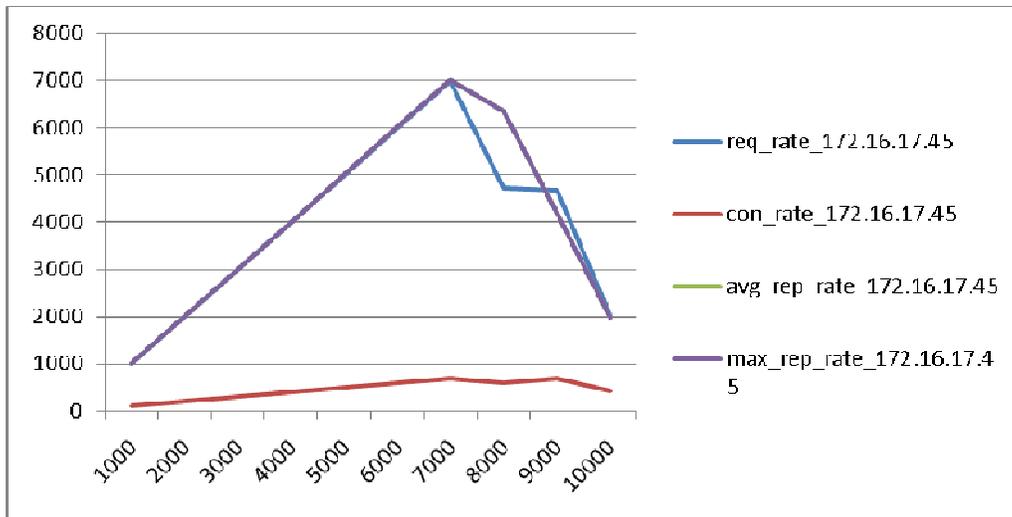
Apache



Sun Java Web Server

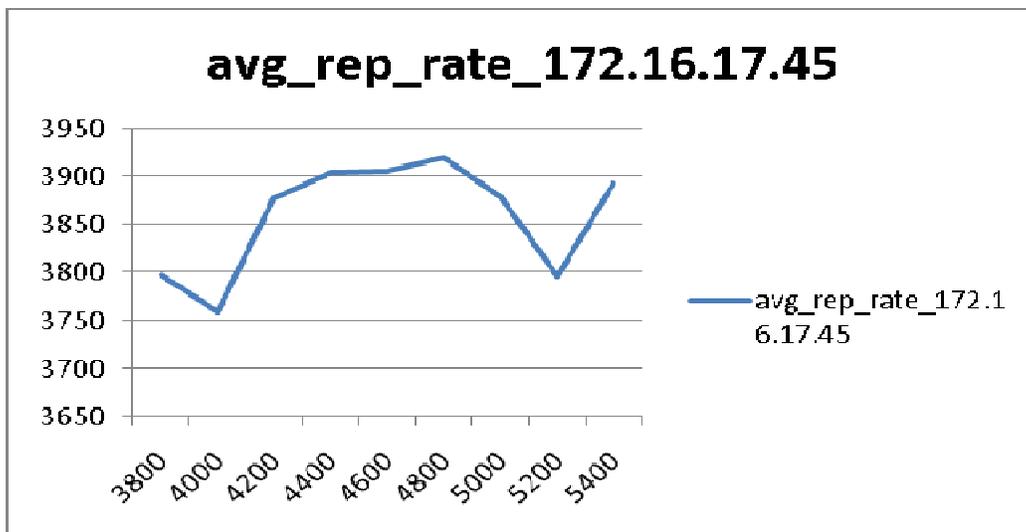


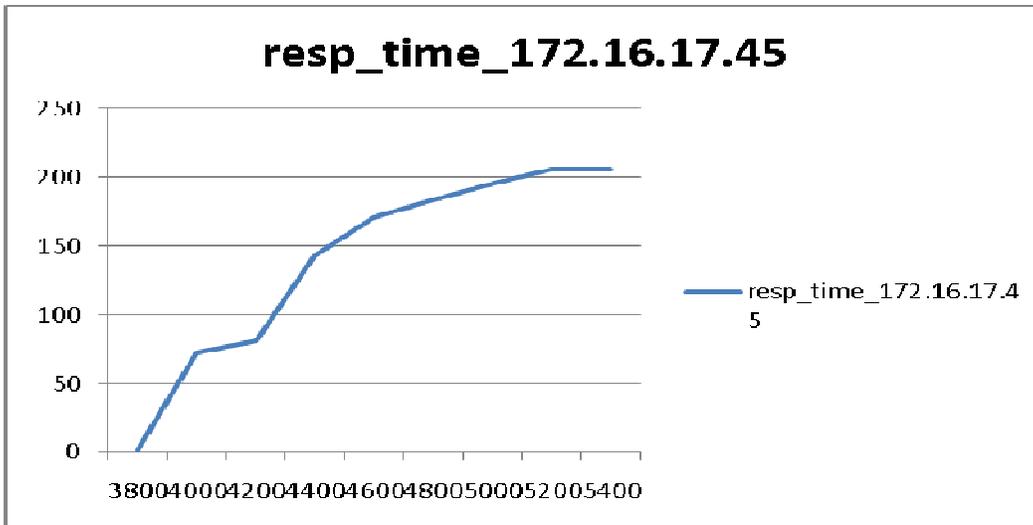
Apache Tomcat



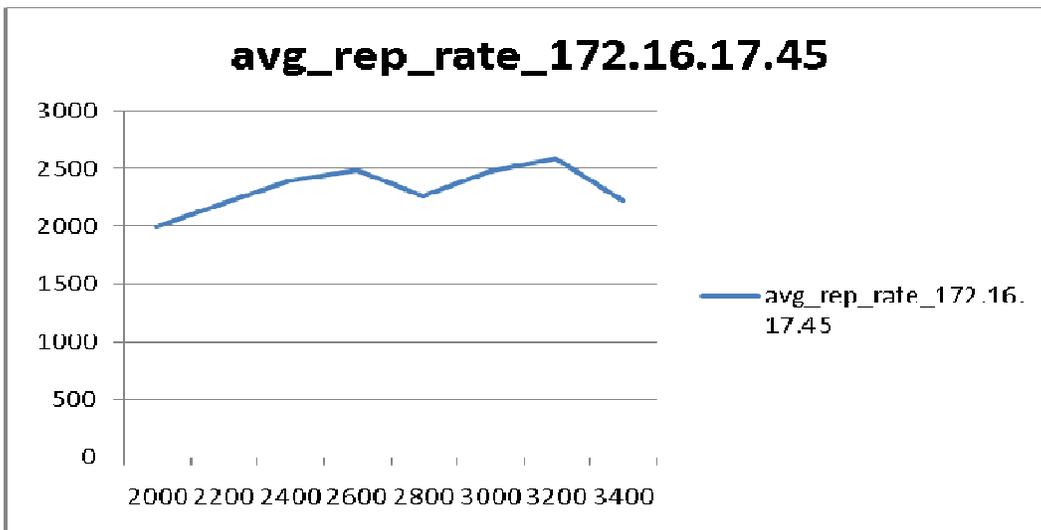
Dynamic pages:

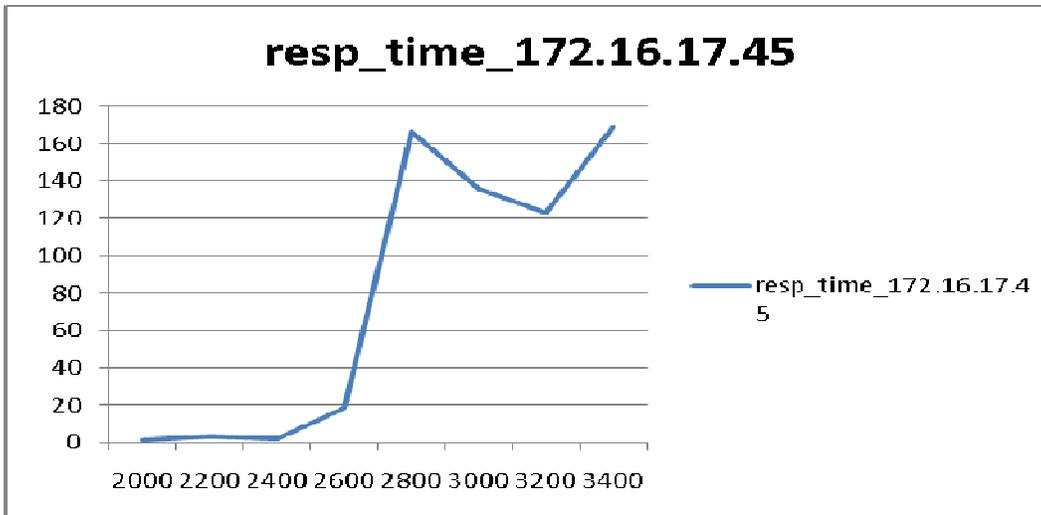
IIS



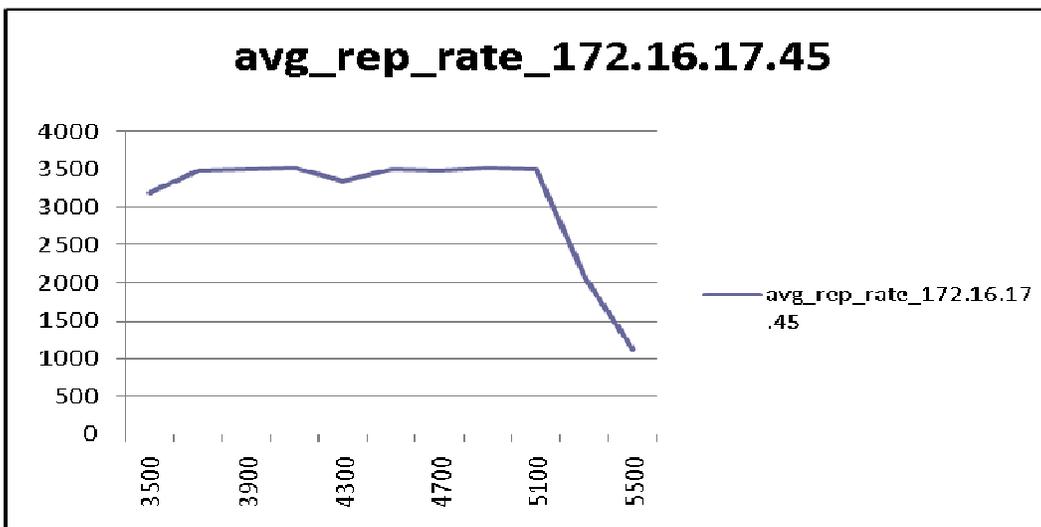


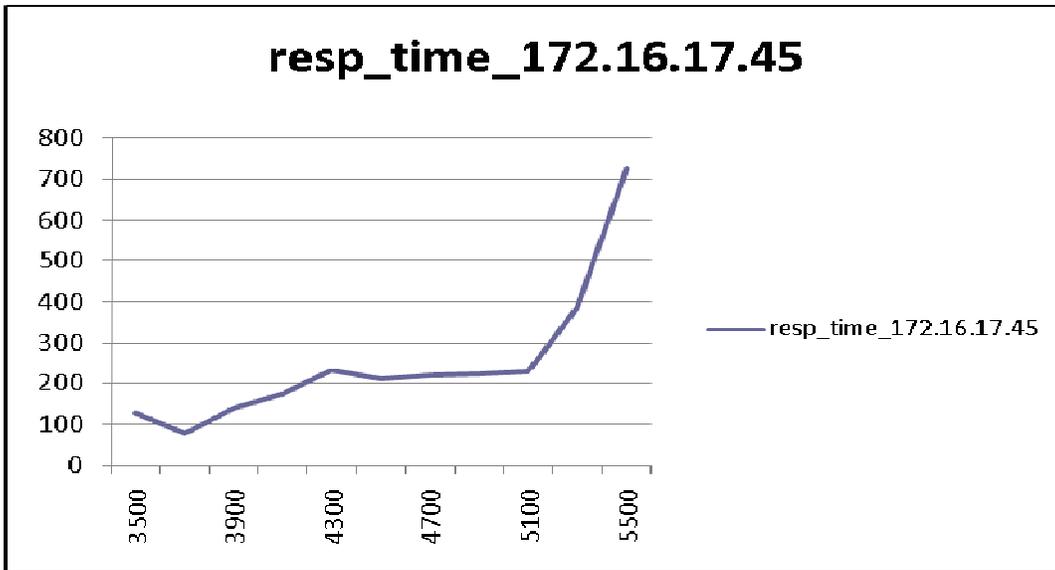
Apache



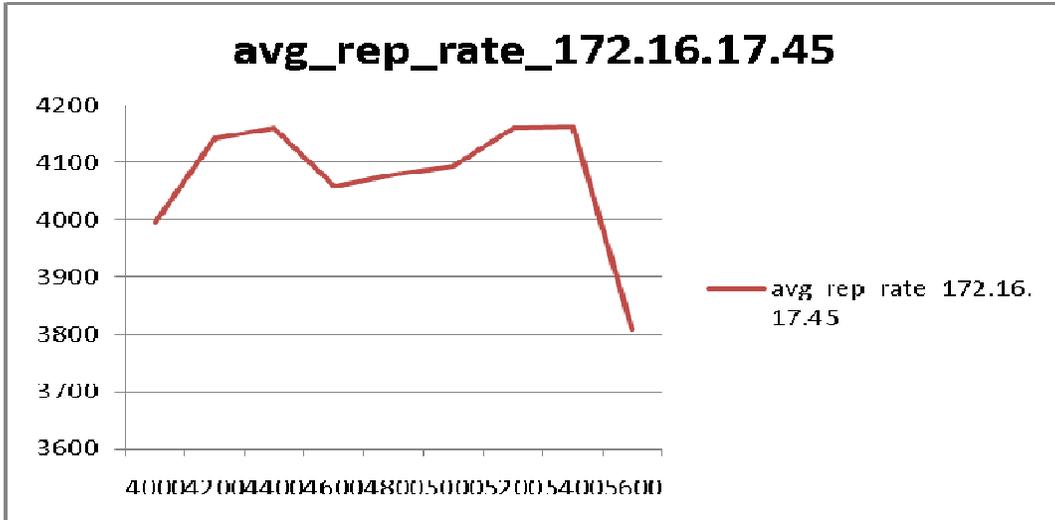


Sun Java Web server





Apache Tomcat



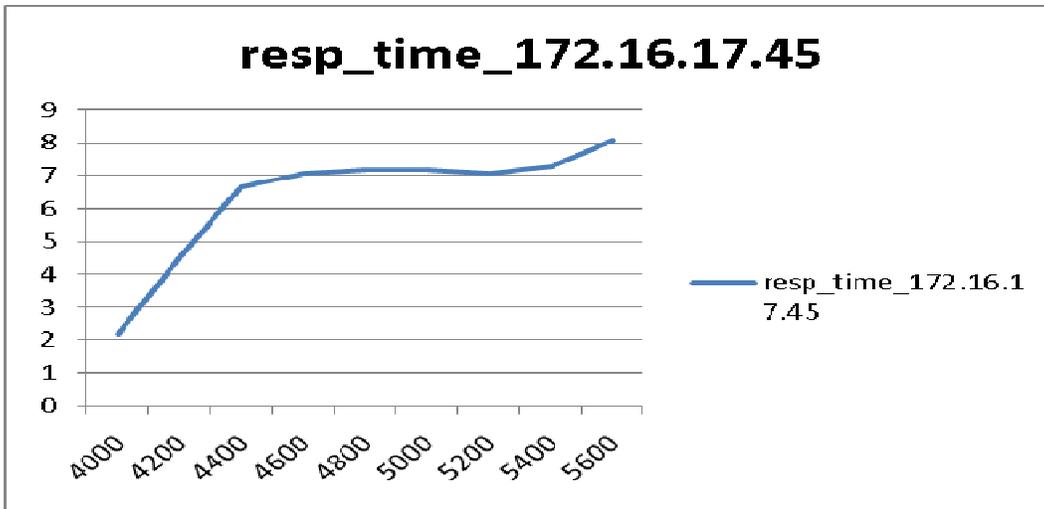


Table showing comparisons of performance

Parameter	IIS	Apache	Tomcat	Sun Java
Overload Point (static page)	6600 req/sec	4000 req/sec	7600 req/sec	6800 req/sec
Average response time (ms)	120 ms	96.8ms	1ms	128 ms
Overload point (dynamic page)	4400 req/sec	2200 req/sec	4600 req/sec	3900 req/sec
Average response time(ms)	142 ms	363.6ms	7ms	139 ms

Conclusion

From the results obtained by testing the various servers using autobench tool, we see that overload/stabilization request rate is maximum for Tomcat. Also, the response time is minimum for Tomcat, (1 ms for static page and 7ms for dynamic page). Sun Java Web Server is next with capability to handle a demand req rate of 6800. The response time at that point is 128 ms. IIS is next with overload at demand request rate of 6600 and the response time at that point being 120 ms.

Apache is next with capability to handle a demand req rate of 4000. The response time at that point is 96.8ms.

6.2 IP Blocking

IIS

Very user friendly. can grant or block particular group / single users using IIS Manager by just inputting the ip of single user or network id + subnet mask of group.

Apache

Gives user the facility of not only blocking the ips but also can hide only some documents in the web server, like blocking only a particular directory. This can be done by having .htaccess file in that particular directory.

Tomcat

Has a built in valve (org.apache.catalina.valves.RemoteAddrValve) for this purpose. The set of IP to be blocked/allowed should be given as regular expressions in the server.xml file.

Sun Java

Not very user friendly. Certain Lines need to be added to the obj.conf file. A WildCard Regular expression representing the blocked ip addresses needs to be added to a built in Server Application Function.

Parameter	IIS	Apache	Tomcat	Sun Java
CPU – utilization (without IP blocked)	13-17%	10-17%	17-24%	15-18%
CPU utilization with IP blocked	9-14%	10-24%	13-19%	20-24%

Memory Usage	16Mb	12Mb	44Mb	98 Mb
--------------	------	------	------	-------

Conclusion

All web servers have IP blocking capability. In Tomcat, the incoming request is processed in a valve and if it is found to be forbidden, then it is sent back with 403 error immediately and no further processing is done for that request while in other servers it is done at some other level during processing.(visible in amount of data sent to user).

6.3 Using SSL Certificate – Self signed , CA signed

IIS

Self – Signed: Created using SelfSSL tool.

CA Signed: Created certificate cer.txt using IIS manager , set up CA using OpenSSL , used OpenSSL tool to get the cer.txt signed by CA. then added the CA certificate to the list of trusted Certificate Authorities.

Apache

Self-Signed: Using openssl.exe present in bin folder in apache.

CA-Signed: Create a keystore with private/public key pair. Export your private key from the keystore to a file server.key, which is your private key. Create a public/private key pair for our sample CA. The CA signs the server certificates. Finally, the CA certificate needs to be imported by the browser.

Tomcat

Self - Signed : Using java keytool and then importing the certificate in the config file(server.xml).

CA signed: Using OpenSSL, a keystore is obtained which is then used by the server. The CA certificate needs to be imported by the browser.

Sun Java

Self-Signed: Can be done Using the Web Based Administration Interface and the command line interface.

CA Signed: Created CSR(Certificate Signing request) using Web based Interface, set up CA used OpenSSL to get the CSR signed by CA,and add CA certificate to the list of trusted Certificate Authorities.

The following table shows performance of each web server when configured with SSL and without SSL.

IIS

With SSL

Threads-rampup (sec)	Throughput (per min)	Average Response time(ms)	Deviation(ms)	Median(ms)
500-3	8576.933	3	4	2
1000-4	11878.836	12	15	5
2000-7	10382.419	74	145	3

Without SSL

Threads-rampup (sec)	Throughput (per min)	Average Response time(ms)	Deviation(ms)	Median(ms)
500-3	8349.569	8	26	2
1000-4	6985.847	191	465	4
2000-7	4365.493	617	920	320

Apache

Without SSL

Threads-rampup (sec)	Throughput (per min)	Average Response time(ms)	Deviation(ms)	Median(ms)
250-2	1095.69	8595	1589	8836
500-4	1690.808	4982	2864	4566
1000-8	1740.493	9216	6177	10123

With SSL

Threads-rampup (sec)	Throughput (per min)	Average Response tim(ms)	Deviation(ms)	Median(ms)
250-2	3602.305	1369	1182	1688
500-4	4401.408	1500	993	910
1000-8	4209.64	2627	2459	1087

Sun Java

Without SSL

Threads-rampup (sec)	Throughput (per min)	Average Response time(ms)	Deviation(ms)	Median(ms)
500-2	11432.003	2	1	3
1000-3	13642.664	3	1	3
2000-3	13336.493	881	706	800

With SSL

Threads-rampup (sec)	Throughput (per min)	Average Response time(ms)	Deviation(ms)	Median(ms)
500-3	8349.569	8	26	2
1000-4	6985.847	191	465	4
2000-7	4365.493	617	920	320

Tomcat

Without SSL

1. The test was run for 500 threads with a ramp up period of 3 seconds.
2. Average response time was always found to be lower than 5ms.
3. Average throughput varied from 7400 to 8700/minute.

With SSL

1. The test was run for 500 threads with a ramp up period of 3 seconds.
2. Average response time was always found to be above 5ms.
3. Average throughput varied from 7000 to 8000/minute.

Conclusion

Self Signed Certificates can easily be installed for all servers.

The Approach to install CA signed certificates is similar in Sun Java Web Server and IIS, which in turn differs from the approach followed by the Apache and Apache Tomcat Web Servers. In the first approach, only the Server can Issue a Certificate Request to a CA, which will create the certificate accordingly and the Certificate can be installed. An external certificate issued without a Certificate signing request cannot be installed, whereas this is not the case for the other approach. This approach followed by IIS and Sun Java Servers is less user friendly and more secure.

6.4 New Extension File handling (writing native server modules)

IIS

A Http Handler to handle files with new extension, create a new C# library using HttpHandler, register the handler by changing web.conf file, add the new extension to isapi mappings.

Apache

A file handler in Apache can be written in various languages like perl, python, c. Also it is dynamically compiled and added to the core system. By default the perl and python modules are not enabled, we need to download, install and configure them with respect to Apache.

Tomcat

Valves can be coded to handle such requests and forward to appropriate servlet to handle it.

Sun Java

A Custom Server Application Function(SAF) in C needs to be written and added as an NSAPI plugin.

Conclusion

New extension file handling can be done with same complexity in all web servers.

6.5 Dynamic Language Support

IIS

IIS is capable of supporting scripting languages including ASP and ASP.net , PHP , CGI scripting.

ISAPI extensions

Need to configure **aspnet_isapi.dll** and **isapi_fcgi.dll** for ASP.net and CGI, PHP files respectively. But in case of JSP, it needs a separate Java based Web Server to process JSP requests and send back the reply. So, IIS doesn't support JSP pages.

Apache

Not only we can use existing CGI programs, in Apache writing our own CGI scripts and integrating is easy too. The Apache also uses SSI (Server Side Includes) to support dynamic content.

Modules allow the embedding of scripting languages into HTML pages. This makes executing the scripts much more efficient, since an interpreter does not need to be started for every request.

Tomcat

Tomcat is capable of supporting JSP, PHP, Servlets, CGI.

ASP was supported through Sun One Active Server Pages, but now the project has entered its end of life phase and is no longer active.

Sun Java

Sun Java is capable of supporting JSP, PHP, Servlets, CGI.

ASP was supported through Sun One Active Server Pages, but now the project has entered its end of life phase and is no longer active.

Conclusion

IIS supports almost all scripting languages except JSP. Sun Java and Tomcat do not support ASP. Apache can support all languages including ASP through a module called mod_mono.

6.6 URL rewriting for hiding extension

IIS

Wrote a HttpModule, which is a dll file got from a C# class library project. Added the dll file to bin directory of extension virtual directory changed the URL of default Http 404 error page to /extension/404.aspx. This above step was required because, it's not possible to call Asp.net for no extensions file in IIS. Change the web.conf file to call the HttpModule.

Apache

The module used for this is mod_rewrite. By default it is not enabled and needs to be enabled in httpd.conf present in conf directory. A condition is to be satisfied by the file whose url needs to be rewritten. A rule is defined according to which the path of the desired file is rewritten.

Tomcat

The pages need to be mapped to other urls using mapping in the web.xml(application descriptor) while developing the application itself.

Sun Java

A built in SAF can be used to map certain URL's to others using wildcards. The SAF is to be added to the obj.conf file.

Conclusion

All web servers have facilities for hiding URLs.

Conclusion

All the four web servers – IIS, Apache, Sun Java and Apache Tomcat have facilities for applications specific with scripting languages. SSL authentication is more secure in Sun Java and IIS. Tomcat stands first in terms of performance. From the evaluation of all the parameters, we conclude that the use of web servers depends on the type of applications hosted on them.