



ESCUELA SUPERIOR POLITECNICA DE CHIMBORAZO
ESCUELA DE INGENIERIA ELECTRONICA
EN TELECOMUNICACIONES Y REDES
“ESTUDIO COMPARATIVO DEL RENDIMIENTO DE
SERVIDORES WEB DE VIRTUALIZACION SOBRE LA
PLATAFORMA WINDOWS SERVER 2008”

TESIS DE GRADO

Previa la obtención del título de:

INGENIERO ELECTRÓNICO, TELECOMUNICACIONES Y REDES

Presentado por:

Jessica Nataly Castillo Fiallos

Riobamba – Ecuador

2012

“...A Dios a mis Padres, mis hermanas, amigos, profesores que me acompañaron durante todo este proceso, no solo en la realización del Proyecto de Grado sino durante toda la carrera.. Gracias por su apoyo incondicional...”

Jessica

*“.....A Dios a mis Padres, mis
hermanas, quienes con abnegación
y mucha paciencia me apoyaron
en todo momento de mi vida.*

Jessica

FIRMAS RESPONSABLES Y NOTA

NOMBRE	FIRMA	FECHA
Ing. Iván Ménes C		
DECANO FACULTAD DE INFORMATICA Y ELECTRONICA	_____	_____
Ing. Pedro Infante M		
DIRECTOR DE ESCUELA ING. EN ELECTRONICA TELECOMUNICACIONES Y REDES	_____	_____
Ing. Santiago Cisneros		
DIRECTOR DE TESIS	_____	_____
Ing. Ivonne Rodríguez		
MIEMBRO DEL TRIBUNAL	_____	_____
Tlgo. Carlos Rodríguez		
DIRECTOR CENTRO DE DOCUMENTACION	_____	_____
NOTA DE LA TESIS: _____		

*“Yo, **JESSICA NATALY CASTILLO FIALLOS**, soy la responsable de las ideas, doctrinas y resultados expuestos en ésta tesis, y el patrimonio intelectual de la misma pertenecen a la ESCUELA SUPERIOR POLITECNICA DE CHIMBORAZO”*

Jessica Castillo Fiallos

ÍNDICE DE ABREVIATURAS

ASP	Active Server Pag
CGI	Common Gateway Interface
DHTML	Dynamic Hyper Text Markup Languaje
DNS	El sistema de nombre de dominios
FTP	File Transfer Protocol (Protocolo de Transferencia de Archivos)
HTML	Hypertext Markup Languaje (Languaje de Marcado de Hipertexto)
HTTP	Hypertext Transfer Protocol (Protocolo de Transferencia de Hipertexto)
IP	Internet Protocol
ISS	Internet Server
JSP	JavaServer Pages, JSP
JVM	Java Virtual Machine.
PHP	Hypertext Preprocessor/Personal
SCGI	Simple Common Gateway Interface
SSI	Server Side Includes
SSL	Secure Socket Layer
URL	Universal Resource Locators
XHTML	Extensible Hypertext Markup Languaje
XML	eXtensible Markup Languaje
WAR	Archivos Web
WWW	World Wide Web

ÍNDICE GENERAL

AGRADECIMIENTO

DEDICATORIA

FIRMAS DE RESPONSABILIDAD

RESPONSABILIDAD DEL AUTOR

ÍNDICE

INTRODUCCIÓN

CAPITULO I

MARCO REFERENCIAL	15
1.1 INTRODUCCION	15
1.2 FORMULACIÓN DEL PROBLEMA	16
1.3 OBJETIVOS.....	18
1.3.1 Objetivo General:.....	18
1.3.2 Objetivos Específicos:.....	18
1.4 JUSTIFICACIÓN DEL PROBLEMA	18
1.5 PLANTEAMIENTO DE LA HIPÓTESIS	20

CAPÍTULO II

MARCO TEÓRICO.....	21
2.1 BREVE HISTORIA DE LA WWW	21
2.2 FUNDAMENTOS DE LA WEB.....	22
2.3 EL PROTOCOLO HTTP	22
2.4 EL LENGUAJE HTML.....	25
2.5 HISTORIA DE LAS APLICACIONES WEB	26
2.5.1 Historia	28
2.5.2 Desarrollo de las Aplicaciones Web	28
2.5.3 Arquitectura Web.....	29
2.6 EL NAVEGADOR WEB (BROWSER).....	30
2.7 EL SERVIDOR HTTP	32
2.7.1 Aplicaciones Multinivel	32
2.7.1.1 Ejemplos de Herramientas de Edición Web.....	35

2.7.2 Conceptos básicos del servidor Web	37
2.8 SERVIDOR WEB APACHE	39
2.8.1 Módulos	41
2.8.2 Uso	41
2.8.3 Configuración	43
2.8.4 Licencia.....	43
2.9 INSTALACIÓN DE APACHE	44
2.9.1 Descargar	44
2.9.2 Inicio de Apache	45
2.10 INTERNET INFORMATION SERVICES (IIS)	46
2.10.1 Historia.....	47
2.10.2 Características.....	48
2.10.3 IIS expreso	50
2.10.4 Extensiones	50
2.10.5 Versiones.....	52
2.11 WINDOWS SERVER 2008.....	53
2.11.1 Características.....	54
2.11.1.1 Más control	54
2.11.1.2 Mayor protección.....	55
2.11.1.3 Mayor flexibilidad	56
2.11.2 VISUALIZACIÓN	57
2.11.2.1 Seguridad	57
2.11.2.2 Fuerte aislamiento	58
2.11.2.3 Rendimiento.....	59
2.11.2.4 Administración simplificada.....	61
2.12 DEFINICIÓN DE MÁQUINA VIRTUAL.....	63
2.12.1 Tipos de máquinas virtuales.....	63
2.12.2Máquinas virtuales de sistema	64
2.12.3Aplicaciones de las máquinas virtuales de sistema	64
2.12.4Máquinas virtuales de proceso	65

2.12.5Inconvenientes de las máquinas virtuales	66
2.12.6Técnicas.....	66
2.12.6.1 Emulación del hardware subyacente (ejecución nativa)	66
2.12.6.2 Emulación de un sistema no nativo	67
2.12.6.3 Virtualización a nivel de sistema operativo	68
2.13 MEDIR EL RENDIMIENTO DESDE UN CLIENTE WINDOWS	69
2.13.1 Colasoft Capsa.....	69
2.13.2 WebServer Stress Tools.....	70
2.13.3 El mensaje a los propietarios de sitios web es clara: probar y monitorear su sitio web.....	71
2.14 MEDIR EL RENDIMIENTO DESDE UN CLIENTE LINUX	73
2.14.1 Comando ab	73
2.14.2 Comando httping	75
2.14.3 Comando httperf.....	76
CAPÍTULO III	
COMPARATIVA DE SERVIDORES.....	79
3.1 INTRODUCCION.....	79
3.2.1 Ambiente de Prueba	81
3.2.3 Paso 1: Identificación de las variables	82
3.2.4 Paso 2: Decidir los métodos y el tiempo de monitoreo.....	86
3.2.5 Paso 4: Definir los sujetos de medición	88
3.2.6 Paso 4: Realizar las Pruebas	88
3.2.6.1 Procedimiento.....	89
3.2.6.2 Configuración de Herramientas para Medición.....	92
3.2.6.2.1 Configuración de Colasoft , Cliente Windows.....	92
a. Configuración de la Tarjeta de Red para Análisis de Tráfico.....	92
b. Configuración de opciones de Red.....	93
c. Configuración del Analizador de Diagnostico.....	93
3.2.6.2.2 Configuración de Web Stress Tools	96
a. Peticiones por Clic	96
b. Peticiones por tiempo con carga constante	98

c. Peticiones por tiempo con carga variable	100
---	-----

CAPITULO IV

ANALISIS DE RESULTADOS	101
4.1 INTRODUCCION	101
4.2 EVALUACIÓN DE LAS CARACTERÍSTICAS	102
4.3 GRAFICAS Y RESULTADOS DE LAS PRUEBAS	104
4.3.1 Prueba 1. Análisis Tráfico de red.....	104
4.3.1.1 PRUEBA 1.1 PING CON CARGA, PING ICMP	112
4.3.2 PRUEBA 2. PRUEBA CON CARGA CONSTANTE POR 2 MINUTOS CON 10 USUARIOS SIMULTÁNEOS.....	113
4.3.3 PRUEBA 3. PRUEBA CARGA VARIABLE POR 2 MINUTOS CON 10 USUARIOS SIMULTÁNEOS	115
4.3.4 PRUEBA 4. LATENCIA, PING IP	117
4.3.4.1 LATENCIA (SIN CARGA).....	118
4.3.4.2 LATENCIA (CON CARGA).....	119
4.3.5 PRUEBA 5. PRUEBA DE SOBRECARGA DE PETICIONES.....	119
4.3.6 PRUEBA 6. PRUEBA DE SOBRECARGA DE USUARIOS	124
4.3.7 PRUEBA 7. PORCENTAJE DE USO DE CPU	126
4.4 ANÁLISIS DE RESULTADOS.....	130
4.5 RESULTADO FINAL DE LA COMPARATIVA	150

CONCLUSIONES

RECOMENDACIONES

RESUMEN

SUMMARY

BIBLIOGRAFIA INTERNET

ANEXOS

ÍNDICE DE FIGURAS

Figura II.1. Funcionamiento del Protocolo Http.....	24
Figura II.2. Arquitectura Web Básica.....	30
Figura II.3. Arquitectura Multinivel.....	34
Figura II.4. Arquitectura Web de tres niveles.....	35
Figura II.5. Logotipo Servidor http Apache.....	39
Figura II.6. Logotipo Internet Information Services.....	46
Figura II.7. Logotipo Windows Server 2008.....	53
Figura II.8. Respuesta de ejecución del comando ab en un cliente Linux	74
Figura II.9. Ejemplo de ejecución comando httping	76
Figura II.10. Respuesta de un servidor web a 30000 peticiones con el comando httpperf..	78
Figura II.11. Diagrama de Flujo para la elaborar la Prueba de Rendimiento.....	81
Figura II.12. Estructura de Red para realizar las pruebas de rendimiento.....	81
Figura III.13. Adaptador para pruebas de tráfico de Red.....	92
Figura III.14. Configuración de Endpoints.....	93
Figura III.15. Configuración de eventos para la capa de Aplicación.....	94
Figura III.16. Configuración de eventos para la capa de Transporte.....	94
Figura III.17. Configuración de eventos para la capa de Red.....	95
Figura III.18. Configuración de eventos para la capa de Enlace de Datos.....	95
Figura III.19. Configuración de peticiones o clicks para estresar los servidores web IIS y Apache.....	97
Figura III.20. Direcciones Web del servidor IIS que serán analizadas.....	98
Figura III.21. Direcciones Web del servidor Apache que serán analizadas.....	98
Figura III.22. Configuración de tiempo para pruebas de carga constante	99
Figura III.22. Configuración de tiempo para pruebas de carga constante	100
Figura IV.24. Tráfico broadcast, multicast y total generado entre el cliente y el servidor IIS.....	104

Figura IV.25. Tráfico broadcast, multicast y total generado entre el cliente y el servidor Apache.....	104
Figura IV.26. Total de Paquetes TCP y Peticiones DNS cliente-servidor IIS.....	105
Figura IV.27. Total de Paquetes TCP y Peticiones DNS cliente-servidor Apache.....	105
Figura IV.28. Peticiones y Conexiones HTTP que se establecen cliente-servidor IIS.....	105
Figura IV.29. Peticiones y Conexiones HTTP que se establecen cliente-servidor Apache.....	106
Figura IV.30. Resultado de errores en peticiones DNS y HTTP (IIS).....	106
Figura IV.31. Resultado de errores en peticiones DNS y HTTP (Apache)	106
Figura IV.32. Análisis de Trafico por Protocolos (IIS)	107
Figura IV.33. Análisis de Trafico por Protocolos (Apache)	107
Figura IV.34. Relación de paquetes con error y paquetes normales intercambiados (ISS).....	108
Figura IV.35. Relación de paquetes con error y paquetes normales intercambiados (Apache).....	108
Figura IV.36. Relación de paquetes broadcast y multicast enviados (ISS).....	109
Figura IV.37. Relación de paquetes broadcast y multicast enviados (Apache).....	110
Figura IV.38. Relación de Conexiones TCP iniciadas y establecidos (ISS).....	110
Figura IV.39. Relación de Conexiones TCP iniciadas y establecidos (Apache).....	110
Figura IV.40. Paquetes HTTP promedio por segundo (ISS).....	111
Figura IV.41. Paquetes HTTP promedio por segundo (Apache).....	111
Figura IV.42. Respuesta de tiempos de ida y vuelta mientras se ejecuta 1000 peticiones por 10 usuarios hacia el servidor ISS.....	112
Figura IV.43. Respuesta de tiempos de ida y vuelta mientras se ejecuta 1000 peticiones por 10 usuarios hacia el servidor Apache	112
Figura IV.44. Promedio de Utilización de Ancho de banda cliente-servidor IIS	113
Figura IV.45. Promedio de Utilización de Ancho de banda cliente-servidor Apache.....	113
Figura IV.46. Memoria del sistema disponible, Trafico de Red y uso del CPU local (ISS).....	114

Figura IV.47. Memoria del sistema disponible, Trafico de Red y uso del CPU local (Apache)	114
Figura IV.48. Ancho de banda promedio utilizado cliente-servidor IIS.....	115
Figura IV.49. Ancho de banda promedio utilizado cliente-servidor Apache.....	115
Figura IV.50. Memoria del sistema disponible, Trafico de Red y uso del CPU local (IIS).....	116
Figura IV.51. Memoria del sistema disponible, Trafico de Red y uso del CPU (Apache).....	116
Figura IV.52. Latencia y tiempos de ida y vuelta, prueba sin carga a.) Servidor IIS, b.) Servidor Apache	118
Figura IV.53. Latencia y tiempos de ida y vuelta, prueba con carga a.) Servidor IIS, b.) Servidor Apache.....	119
Figura IV.54. Número de peticiones por segundo y tiempo por cada petición que soporta sin carga a. Servidor IIS y b. Servidor Apache.....	121
Figura IV.55. Número de peticiones por segundo y tiempo por cada petición que soporta con carga a. Servidor IIS y b. Servidor Apache	124
Figura IV.56. Resultado sin éxito de sobre carga de usuarios para el servidor IIS	124
Figura IV.57. Resultado con éxito de sobre carga de usuarios para el servidor IIS	125
Figura IV.58. Porcentaje de uso de CPU con prueba sin carga cliente-servidor para a. Servidor IIS y b. Servidor Apache.....	127
Figura IV.59. Porcentaje de uso de CPU, prueba con carga cliente-servidor para a. Servidor IIS y b. Servidor Apache	129
Figura IV.60. Grafica porcentual de las características de los servidores.....	148
Figura IV.61. Grafica porcentual del rendimiento de los servidores y recursos del Sistema.....	149
Figura IV.62. Grafica porcentual del Resultado Final de la Comparativa.....	150

ÍNDICE DE TABLAS

Tabla III.I.- Definición de variables a medir, Análisis de Tráfico	83
Tabla III.II. Definición de variables a medir, Análisis de Rendimiento	85
Tabla III.III.- Definición de variables a medir, Análisis de Tiempos	85
Tabla III.IV. Características Técnicas del sujeto sometido a rueba.....	88
Tabla III.V. Umbrales recomendados de rendimiento para su sitio web o aplicación web.....	90
Tabla IV.VI. Características de cada servidor	102
Tabla IV.VII. Descripción de Asignación de pesos	130
Tabla IV.VIII. Resultados Prueba 1. Estrés, 10 usuarios 1000 peticiones por usuario ...	131
Tabla IV.IX Asignación de pesos para obtener el servidor apropiado a nuestra necesidad Prueba 1	134
Tabla IV.X. Resultados Prueba2. Carga Constante por 2 minutos con 10 usuarios simultáneos	136
Tabla IV.XI. Resultados Prueba 2	137
Tabla IV.XII. Resultados Prueba3. Carga Variable por 2 minutos con 10 usuarios simultáneos	138
Tabla IV.XIII. Resultados Prueba 3	139
Tabla IV.XIV. Resultados Prueba 4. Latencia usando ping IP, httping	140
Tabla IV.XV. Resultados Prueba 4	141
Tabla IV.XVI. Resultados Prueba 5. Sobre carga de peticiones	142
Tabla IV.XVII. Resultados Prueba 5	143
Tabla IV.XVIII. Resultados Prueba 6. Sobre carga de usuarios	144
Tabla IV.XIX. Resultados Prueba 6	145
Tabla IV.XX. Resultados Prueba 7. Porcentaje de uso del CPU.....	146
Tabla IV.XXI. Resultados Prueba 7.....	146
Tabla IV.XXII. Resultados de las Pruebas	147
Tabla IV.XXIII. Resultado Final	149

CAPÍTULO I

MARCO REFERENCIAL

1.1 INTRODUCCION

Debido al auge que se encuentra viviendo el mundo entorno a la comunicación mediante el internet, el florecimiento de la posibilidad irrestricta de compartir información mediante los sitios de la plataforma Web, se suscita una problemática que aún su discusión no se ha tomado de manera que se debe. Los sitios Web son de por si un medio de comunicación de masas, que pueden influir, educar o informar de una manera casi gratuita por esta razón se va desarrollando rápidamente la virtualización ya que así se utilizan de mejor manera los recursos y el rendimiento de los diferentes procesos de cada uno de los servidores.

En la presente investigación se realizará una comparativa de virtualización de Servidores Web, Apache e ISS sobre la plataforma de Windows Server 2008, tomando como parámetros de comparación sus características implementadas, pruebas de rendimientos, utilización de recursos, así como también sus fortalezas y debilidades. En base al análisis comparativo y a un conjunto de requerimientos necesarios para las pruebas de carga en cada servidor Web, lo cual permitirá saber cuál es el más óptimo y el que va a cumplir la hipótesis planteada.

1.2 FORMULACIÓN DEL PROBLEMA

En la actualidad Internet se ha convertido en una herramienta necesaria para todas las personas ya que nos permite realizar diferentes actividades gracias a la difusión en incremento de sitios Web en los últimos años. En estos días los términos correo electrónico, foros de discusión, tiendas virtuales, etc. Son muy comunes en nuestra sociedad y nos han hecho experimentar cambios significativos en el concepto que se tenía anteriormente de una computadora.

Ahora lo que se busca es la mejor utilización de los recursos por eso existen muchas razones para la adopción de la virtualización de servidores. Una muy popular es la mejor utilización de recursos. No es raro ver a los servidores que ejecutan el diez por ciento o menos de su capacidad, en diferentes puntos en el día. Al permitir que compartan varios servidores virtuales de un único conjunto de

hardware, un índice mucho más alto promedio de utilización se logra, y los costos de hardware y soporte bajan.

La virtualización también facilita el suministro y reasignar los servidores. En lugar de tener que configurar manualmente un servidor, el software de virtualización puede configurar un servidor utilizando una plantilla pre-existentes y las imágenes de cambio de servidor de un servidor físico a otro para equilibrar cargas de trabajo o mejorar la eficiencia. También puede configura automáticamente un nuevo servidor virtual en un equipo diferente cuando hay un mal funcionamiento del hardware. Cada aplicación está aislada de los demás, que proporciona una mayor seguridad.

Un sistema operativo que soporta este tipo de tecnología es Windows Server 2008. Así en esta investigación se utilizará software libre para reducir costos.

Dada la necesidad presentada de la mejor utilización de recursos se propone un Estudio Comparativo del rendimiento de servidores Web de virtualización sobre la plataforma de Windows Server 2008 con el objetivo de comparar el rendimiento de servidores virtuales de páginas Web, Apache e ISS para la elección del más eficiente.

Para llevar a cabo este estudio se realizarán pruebas de carga a los servidores Web motivos del estudio, Apache Http Server, ISS, sobre la plataforma Windows Server 2008, evaluándose:

1.3 OBJETIVOS

1.3.1 Objetivo General:

“Realizar un estudio comparativo entre herramientas de virtualización para servidores IIS y APACHE sobre la plataforma Windows Server 2008.”

1.3.2 Objetivos Específicos:

- Hacer uso del software indicado que permita analizar el rendimiento de cada servidor.
- Definir los parámetros y ambiente de simulación.
- Realizar un estudio comparativo del rendimiento de virtualización de servidores web.

1.4 JUSTIFICACIÓN DEL PROBLEMA

En la actualidad existen muchas razones para la adopción de la virtualización de servidores. Una muy popular es la mejor utilización de recursos. No es raro ver a los servidores que ejecutan el diez por ciento o menos de su capacidad, en diferentes puntos en el día. Al permitir que compartan varios servidores virtuales

de un único conjunto de hardware, un índice mucho más alto promedio de utilización se logra, y los costos de hardware y soporte bajan.

La virtualización también facilita el suministro y reasignar los servidores. En lugar de tener que configurar manualmente un servidor, el software de virtualización puede configurar un servidor utilizando una plantilla pre-existentes y las imágenes de cambio de servidor de un servidor físico a otro para equilibrar cargas de trabajo o mejorar la eficiencia. También puede configura automáticamente un nuevo servidor virtual en un equipo diferente cuando hay un mal funcionamiento del hardware. Cada aplicación está aislada de los demás, que proporciona una mayor seguridad.

Internet Information Services, IIS, es una serie de servicios para los ordenadores que funcionan con Windows. El servidor Web se basa en varios módulos que le dan capacidad para procesar distintos tipos de páginas

APACHE es el servidor de páginas web por excelencia. Es un servidor rápido y estable y no excesivamente complejo de configurar, además tiene en su apoyo una gran cantidad de información, sobre sus funcionalidades en Internet. Tiene una gran cantidad de módulos que podemos instalar para ampliar sus funcionalidades.

IIS Y APACHE se destacan por su amplia utilización y ventajas mostradas frente a otros servidores. Windows Server 2008 es un sistema operativo de Microsoft diseñado para servidor. La implementación de software a menudo es una tarea

planificada que requiere tiempo de desarrollo por lo cual se hace indispensable el uso de herramientas eficientes como Linux.

Mediante el estudio comparativo de los servidores; IIS Y APACHE se logra reconocer ventajas y desventajas de la virtualización de servidores; teniendo esto diversas aplicaciones, en nuestro caso centraremos el estudio a la comparación de herramientas de la virtualización de servidores.

El presente trabajo desarrollará un ambiente de simulación; el cual se la realizara para las respectivas pruebas y comparaciones.

La implementación de la aplicación demostrativa se la llevará a cabo sobre el servidor Windows server 2008 por su robustez y soporte a todo tipo de tecnologías.

1.5 PLANTEAMIENTO DE LA HIPÓTESIS

“El estudio comparativo del rendimiento de virtualización de servidores APACHE e IIS, conllevará a la selección de la más adecuada”.

CAPÍTULO II

MARCO TEÓRICO

2.1 BREVE HISTORIA DE LA WWW

World Wide Web (o la Web) o Red Global Mundial es un sistema de documentos de hipertexto y/o hipermedios enlazados y accesibles a través de internet. Con un navegador Web, un usuario visualiza páginas Web que puedan contener texto, imágenes, videos u otros contenidos multimedia, y navega a través de ellas usando hiperenlaces.

La Web fue creada alrededor de 1990 por el Inglés Tim Berners-Lee y el belga Robert Calliau mientras trabajaban en el CERN (Centro Europeo de Investigación Nuclear) en Ginebra. Suiza. Desde entonces, Berners-Lee ha jugado un papel

activo guiando el desarrollo de estándares Web (como los lenguajes de marcado con los que se crean las páginas Web), y en los últimos años ha abogado por su visión de una Web Semántica.

2.2 FUNDAMENTOS DE LA WEB

El éxito espectacular de la Web se basa en dos puntuales fundamentos: el protocolo HTTP y el lenguaje HTML. Uno permite una implementación simple y sencilla de un sistema de comunicaciones que nos permite enviar cualquier tipo de ficheros de una forma fácil, que simplifica el funcionamiento del servidor, permitiendo que servidores poco potentes atiendan miles de peticiones y simplificando los costes de despliegue. El otro nos proporciona un mecanismo de composición de páginas enlazadas simple y fácil, altamente eficiente y de uso muy simple.

2.3 EL PROTOCOLO HTTP

El Protocolo HTTP (Hypertext Transfer Protocol) es un sencillo protocolo cliente-servidor que articula los intercambios de información entre los clientes Web y los servidores HTTP. La especificación completa del protocolo HTTP 1/0 está recogida en el RFC 1945. Fue propuesto por Tim Berners-Lee, atendiendo a las

necesidades de un sistema global de distribución de información como el WWW (World Wide Web).

Desde el punto de vista de las comunicaciones, está soportado sobre los servicios de conexión TCP/IP, y funciona de la misma forma que el resto de los servicios comunes de los entornos UNIX: un proceso servidor escucha en un puerto de comunicaciones TCP (por defecto, el 80), y espera las solicitudes de conexión de los clientes Web. Una vez que se establece la conexión, el protocolo TCP se encarga de mantener la comunicación y garantizar un intercambio de datos libre de errores.

El protocolo no mantiene estado, es decir, cada transferencia de datos es una conexión independiente de la anterior, no manteniendo ninguna relación entre ellas. Esto es así hasta el punto de que para transferir una página Web debemos enviar el código HTML de texto así como las imágenes que la componen, pues en la especificación inicial de HTTP, la 1.0, se abrían y usaban tantas conexiones como componentes tenía la página, transfiriéndose por cada conexión un componente (el texto de la página o cada una de las imágenes).

Existe una variable de HTTP llamada HTTPS (S por secure) que utiliza el protocolo de seguridad SSL (Secure Socket Layer) para cifrar y autenticar el tráfico entre el cliente y servidor, siendo esta muy usada por los servidores Web de comercio electrónico o que contengan información personal o confidencial.

El funcionamiento esquemático de HTTP es el siguiente; el cliente establece una conexión TCP hacia el servidor, hacia el puerto HTTP (o el indicado en la conexión), envía un comando HTTP de petición de un recurso (junto con algunas cabeceras informativas) y por la misma conexión el servidor responde con los datos solicitados así como algunas cabeceras informativas.

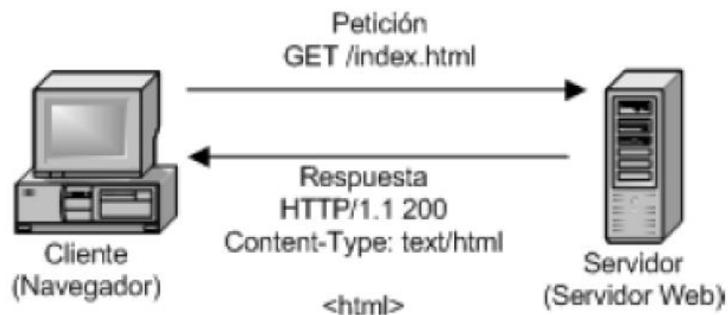


Figura II.1. Funcionamiento del Protocolo Http

El protocolo define además como codificar el paso de parámetros entre páginas, el tunelizar las conexiones (para sistemas de firewall), define la existencia de servidores intermedios de cache, etc.

Las directivas de petición de información que tiene HTTP 1.1 (la versión considerada estable y al uso) son:

GET, Petición de recurso.

POST, Petición de recurso pasando parámetros.

HEAD, Petición de datos sobre recurso.

PUT, Creación o envío de recurso.

DELETE, Eliminación de recurso.

TRACE, Devuelve al origen la petición tal como se ha recibido en el receptor, para depurar errores.

OPTIONS, Sirve para comprobar las capacidades del servidor.

CONNECT, Reservado para uso en servidores intermedios capaces de funcionar como túneles.

Detallaremos a continuación algunos de estos comandos, ya que su comprensión es fundamental para el desarrollo de aplicaciones Web.

Destacar que todos los recursos a ser servidores mediante HTTP deberían ser referenciados mediante una URL (Universal Resource Locators).

2.4 EL LENGUAJE HTML

El código de las páginas está escrito en un lenguaje llamado HTML, que indica básicamente donde colocar cada texto, cada imagen o cada video y la forma que tendrán estos al ser colocados en la página.

El HTML se creó en un principio con objetivos divulgativos. No se pensó que la web llegara a ser un área de ocio con carácter multimedia, de modo que, el HTML

se creó sin dar respuesta a todos los posibles usos que se le iba a dar y a todos los colectivos de gente que lo utilizarían en un futuro.

El lenguaje HTML actualmente se encuentra en la versión 4.01 y empieza a proporcionar funcionalidades más avanzadas para crear páginas más ricas en contenido. Además se ha definido una especificación compatible con HTML, el XHTML (Extensible Hypertext Markup Language) que se suele definir como una versión XML validable de HTML, proporcionándonos un XML Schema contra el que validar el documento para comprobar si está bien formado, etc.

2.5 HISTORIA DE LAS APLICACIONES WEB

En un principio la web era sencillamente una colección de páginas estáticas, documentos, etc., para su consulta o descarga. El paso inmediatamente posterior en su evolución fue la inclusión de un método para elaborar páginas dinámicas que permitieran que lo mostrado tuviese carácter dinámico (es decir, generado a partir de los datos de la petición). Este método fue conocido como CGI (Common Gateway Interface) y definía un mecanismo mediante el que se podía pasar información entre el servidor y ciertos programas externos. Los CGIs siguen utilizándose ampliamente; la mayoría de los servidores web permiten su uso debido a su sencillez. Además, dan total libertad para elegir el lenguaje de programación que se desea emplear.

El funcionamiento de los CGIs tenía un punto débil: cada vez que se recibía una petición, el servidor debía lanzar un proceso para ejecutar el programa CGI. Como la mayoría de CGIs estaban escritos en lenguajes interpretados, como Perl, o en lenguajes que requerían "run-time environment", como Java o VisualBasic, el servidor se veía sometido a una gran carga. La concurrencia de múltiples accesos al CGI podía comportar problemas graves.

Por eso se empiezan a desarrollar alternativas a los CGIs que solucionaran el problema del rendimiento. Las soluciones llegan básicamente por 2 vías: 1) se diseñan sistemas de ejecución de módulos mejor integrados con el servidor, que evitan la instanciación y ejecución de varios programas, y 2) se dota a los servidores un intérprete de algún lenguaje de programación que permita incluir el código en las páginas de forma que lo ejecute el servidor, reduciendo el intervalo de respuesta.

Entonces se experimenta un aumento del número de arquitecturas y lenguajes que permiten desarrollar aplicaciones web. Todas siguen alguna de estas vías. Las más útiles y las más utilizadas son las que permiten mezclar los 2 sistemas: un lenguaje integrado que permita al servidor interpretar comandos "incrustados" en las páginas HTML y, además, un sistema de ejecución de programas mejor enlazado con el servidor, que no implique los problemas de rendimiento propios de los CGIs.

2.5.1 Historia

En los primeros tiempos de la computación cliente-servidor, cada aplicación tenía su propio programa cliente y su interfaz de usuario, estos tenían que ser instalados separadamente en cada estación de trabajo de los usuarios. Una mejora al servidor, como parte de la aplicación, requería típicamente una mejora de los clientes instalados en cada una de las estaciones de trabajo, añadiendo un costo de soporte técnico y disminuyendo la eficiencia del personal.

En contraste, las aplicaciones Web generan dinámicamente una serie de páginas en un formato estándar, soportado por navegadores Web comunes como HTML o XHTML. Se utilizan lenguajes interpretados, del lado del cliente, tales como JavaScript, para añadir elementos dinámicos a la interfaz de usuario. Generalmente cada página Web individualmente es enviada al cliente como un documento estático, pero la secuencia de páginas provee de una experiencia interactiva.

2.5.2 Desarrollo de las Aplicaciones Web

Con la introducción de Internet y del Web en concreto, se han abierto infinidad de posibilidades en cuanto al acceso a la información desde casi cualquier sitio. Esto representa un desafío a los desarrollos de aplicaciones, ya que los avances en

tecnología demandan cada vez aplicaciones más rápidas, ligeras y robustas que permiten utilizar la Web.

El viejo CGI ha cumplido con el propósito de añadir interactividad a las páginas Web pero sus deficiencias en el desarrollo de aplicaciones y en la escalabilidad de las mismas ha conducido al desarrollo de APIs específicos de servidor como Active Server Page, ASP, y PHP, que son más eficientes que su predecesor CGI.

Para aprovechar el potencial de estas tecnologías y ofertar una solución de servidor más extensible y portable. Los servidores Java son muy eficientes, debido al esquema de threads en el que se basan y al uso de una arquitectura estándar como la JVM, Java Virtual Machine.

Otra nueva tecnología viene a sumarse a las que extienden la funcionalidad de los servidores Web, llamada JavaServer Pages, JSP. Los JSP permiten unir HTML, aplicaciones Java, y componentes como las JavaBeans creando una página Web especial que el servidor Web compila dinámicamente en un servidor la primera vez que es llamada.

2.5.3 Arquitectura Web

Ante tal aluvión de posibilidades, conviene repasar aspectos básicos de la arquitectura Web. Para abrir una página Web en un navegador, normalmente se

teclea el correspondiente URL o se pica en el hiperenlace oportuno. Una vez que se solicita esta petición mediante el protocolo HTTP y la recibe el servidor Web, ésta localizada la página Web en su sistema de ficheros y la envía de vuelta al navegador que le solicito, según se muestra en la ilustración II.3. Arquitectura Web básica.



Figura II.2.Arquitectura Web Básica

2.6 EL NAVEGADOR WEB (BROWSER)

El navegador puede considerarse como una interfaz de usuario universal. Dentro de sus funciones están la petición de las páginas Web, la representación adecuada de sus contenidos y la gestión de los posibles errores que se puedan producir.

Para todo esto, los fabricantes de navegadores les han dotado de posibilidades de ejecución de programas de tipo script, con modelos de objetos que permiten manipular los contenidos de los documentos. Estos lenguajes de programación son VBScript, JScript (ambas de Microsoft) y JavaScript (de Netscape), y proporcionan las soluciones llamadas del lado del cliente, client side y permiten realizar validaciones de datos recogidos en las páginas antes de enviarlos al servidor y proporcionan un alto grado de interacción con el usuario dentro del documento.

Otras de las posibilidades de los navegadores es la gestión del llamado HTML dinámico (Dinamic HTML, DHTML). Éste está compuesto de HTML, hojas de estilo en cascada, (Cascade Style Sheets, CSS), modelo de objetos y scripts de programación que permiten formatear y posicionar correctamente los distintos elementos HTML de las páginas Web, permitiendo un mayor control sobre la visualización de las páginas.

En esta línea los navegadores has ido un poco más allá y permiten las visualizaciones de documentos XML (eXtensible Markup Languaje) después de haber sido transformado adecuadamente a HTML por las hojas de estilo extensibles (eXtensible Style Sheets, XSL). De esta manera se puede elegir visualizar ciertos elementos y otros no, dependiendo de las circunstancias.

Además, los navegadores permiten la ejecución de aplicaciones dentro de los documentos mostrados. Las dos posibilidades más populares son la tecnología

ActiveX y los applets Java. Los applets Java son pequeños programas que se descargan del servidor Web y se ejecutan en el JVM del navegador.

2.7 EL SERVIDOR HTTP

El servidor http o servidor Web es un programa que corre el servidor que escucha las peticiones HTTP que le llegan y las satisface. Dependiendo del tipo de la petición, el servidor Web buscará una página Web o bien ejecutará un programa en el servidor. De cualquier modo, siempre devolverá algún tipo de resultado HTML al cliente o navegador que realizó la petición.

El servidor Web va a ser fundamental en el desarrollo de las aplicaciones del lado del servidor, ya que se ejecutará sobre él.

2.7.1 Aplicaciones Multinivel

Al hablar del desarrollo de aplicaciones Web resulta adecuado presentarlas dentro de las aplicaciones multinivel. Los sistemas típicos cliente/servidor pertenecen a la categoría de las aplicaciones de dos niveles. La aplicación reside en el cliente mientras que la base de datos se encuentra en el servidor. En este tipo de aplicaciones el peso del cálculo recae en el cliente, mientras que el servidor hace la parte menos pesada, y eso que los clientes suelen ser máquinas menos

potentes que los servidores. Además, está el problema de la actualización y el mantenimiento de las aplicaciones, ya que las modificaciones a la misma han de ser trasladada a todos los clientes.

Para solucionar estos problemas se ha desarrollado el concepto de arquitecturas de tres niveles: Interfaz de presentación, lógica de la aplicación y los datos.

La capa intermedia es el código que el usuario invoca para recuperar los datos deseados. La capa de presentación recibe los datos y los formatea para mostrarlos adecuadamente. Esta división entre la capa de presentación y la de la lógica permite una gran flexibilidad a la hora de construir aplicaciones, ya que se pueden tener múltiples interfaces sin cambiar la lógica de la aplicación.

La tercera capa consiste en los datos que gestiona la aplicación. Estos datos pueden ser cualquier fuente de información como una base de datos o documentos XML.

Convertir un sistema de tres niveles a otro multinivel es fácil ya que consiste en extender la capa intermedia permitiendo que convivan múltiples aplicaciones en lugar de una sola (véase la Ilustración II. Arquitectura Multinivel).

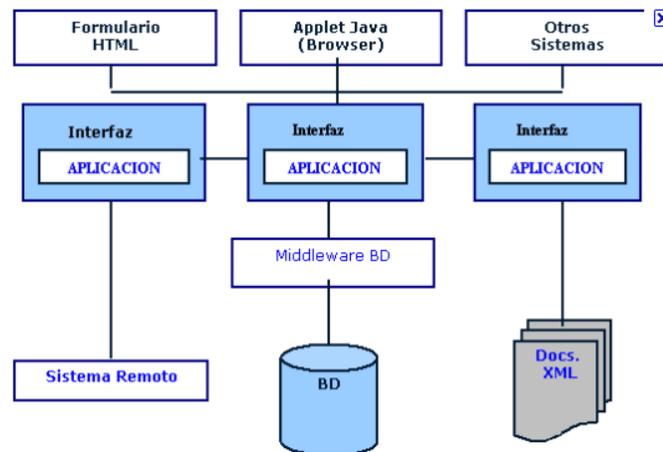


Figura 3.3. Arquitectura Multinivel

La arquitectura de las aplicaciones Web suelen presentar un esquema de tres niveles. El primer nivel consiste en la capa de presentación que incluye no sólo el navegador, sino también el servidor Web que es el responsable de dar a los datos un formato adecuado. El segundo nivel está referido habitualmente a algún tipo de programa o script. Finalmente, el tercer nivel proporciona al segundo los datos necesarios para su ejecución.

Una aplicación Web típica recogerá datos del usuario (primer nivel), los enviará al servidor, que ejecutará un programa (segundo y tercer nivel) y cuyo resultado será formateado y presentado al usuario en el navegador (primer nivel otra vez).

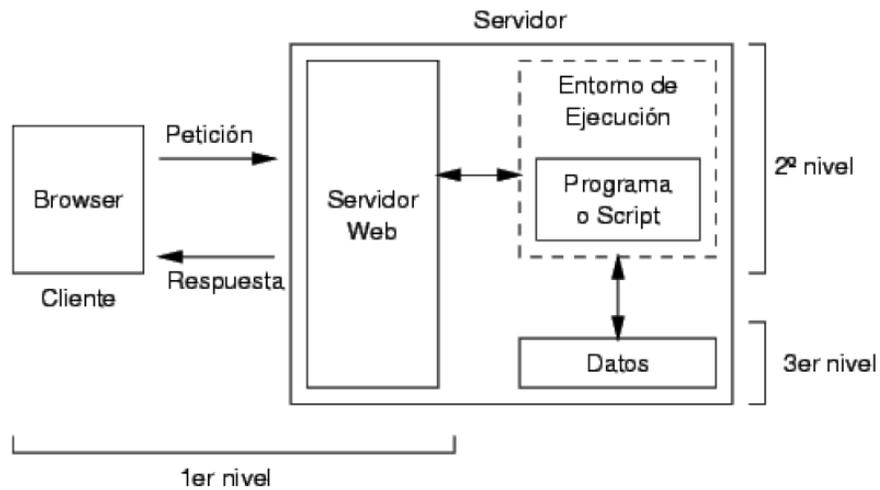


Figura II.4.Arquitectura Web de tres niveles

2.7.1.1 Ejemplos de Herramientas de Edición Web

Existen multitud de herramientas que permiten desarrollar portales Web con aspecto y prestaciones profesionales, aunque la mayoría de ellos requiere de una fase inicial de entrenamiento de duración variable según los casos.

Entre los más utilizados están:

- Macromedia (DreamWeaver, Flash)
- Microsoft FrontPage
- Microsoft Studio .NET en todas las versiones conocidas
- PHP
- MapServer

De los tres primeros hay que decir que integran en un entorno amigable casi todos los elementos relacionados con el desarrollo de páginas Web, además de aportar soluciones a la gestión y el mantenimiento de sitios Web grandes y complejos, y la interacción con bases de datos, El problema en su uso es que no resultan fáciles de manejar y requieren ciertos conocimientos y son de pago.

El tercer caso es el editor de páginas Web que proporciona el navegador Netscape. No es más que un pequeño editor de páginas sin más posibilidades, pero lo fácil de su uso hace que sea una buena herramienta para iniciarse en la creación y en el mantenimiento de portales Web. El precio a pagar es que algunas cosas se han de seguir realizando a mano, como la conexión con bases de datos y la gestión de formularios.

El cuarto caso es PHP que se lo encuentra tanto para Linux como Windows y nos permite crear páginas Web sencillas, dinámicas y con acceso a Base de Datos, y MapServer es un ambiente de desarrollo de código abierto para construir aplicaciones Web espaciales construido sobre otros sistemas de código abierto o freeware y que corre tanto bajo plataformas UNIX/Linux como sobre plataforma Windows 95 o superior.

2.7.2 Conceptos básicos del servidor Web

En informática, un servidor es un tipo de software que realiza ciertas tareas en nombre de los usuarios. Servicios habituales son los servicios de archivos, que permiten a los usuarios almacenar y acceder a los archivos de un ordenador y los servicios de aplicaciones, que realizan tareas en beneficio directo del usuario final. Este es el significado original del término.

Es posible que un ordenador cumpla simultáneamente las funciones de cliente y de servidor. El término servidor ahora también se utiliza para referirse al ordenador físico en el cual funciona ese software, una máquina cuyo propósito es proveer datos de modo que otras máquinas puedan utilizar.

Este uso dual puede llevar a confusión. Por ejemplo, en el caso de un servidor Web, este término podría referirse a la máquina que almacena y maneja los sitios Web, y en este sentido es utilizada por las compañías que ofrecen hosting u hospedaje. Los archivos para cada sitio de Internet se almacenan y se ejecutan en el servidor. Hay muchos servidores en Internet y muchos tipos de servidores, pero comparten la función común de proporcionar el acceso a los archivos y servicios. Un servidor sirve información a los ordenadores que se conecten a él.

Cuando los usuarios se conectan a un servidor pueden acceder a programas, archivos y otra información del servidor.

Los servidores Web, servidores de correo y servidores de bases de datos son a lo que tiene acceso la mayoría de la gente al usar Internet. Algunos servidores manejan solamente correo o solamente archivos, mientras que otros hacen más de un trabajo, ya que un mismo ordenador puede tener diferentes programas de servidor funcionando al mismo tiempo.

Los servidores se conectan a la red mediante una interfaz que puede ser una red verdadera o mediante conexión vía línea telefónica o digital.

Un servidor Web es un programa que atiende y responde las diversas peticiones que le realizan los navegadores, proporcionándoles los recursos que solicitan mediante el protocolo HTTP o el protocolo HTTPS (la versión segura, cifrada y autenticada, de HTTP). Un servidor

Web básico tiene un esquema de funcionamiento muy sencillo, ejecutando de forma infinita el bucle siguiente:

1. Espera peticiones en el puerto TCP asignado (el estándar para HTTP es el 80).
2. Recibe una petición.
3. Busca el recurso en la cadena de petición.
4. Envía el recurso por la misma conexión por donde ha recibido la petición.
5. Vuelve al punto 2.

Un servidor Web que siguiese el esquema anterior cumpliría los requisitos básicos de los servidores HTTP, aunque, eso sí, sólo podría servir ficheros estáticos.

A partir del esquema anterior se han diseñado y construido todos los programas servidores de HTTP que existen, variando sólo qué tipo de peticiones (páginas estáticas, CGI, Servlets, etc.) pueden atender, en si son o no multi-proceso, multi-hilados, etc.

A continuación detallaremos algunas de las características principales de los servidores Web, que extienden, obviamente el esquema anterior.

2.8 SERVIDOR WEB APACHE



Figura II.5. Logotipo Servidor http Apache

Hoy en día, el servidor Web Apache es el servidor más usado de Internet, con una utilización del 67% aproximadamente.

La primera aparición de Apache fue en Abril de 1995. Este servidor se sigue desarrollando “en Internet” como un proyecto de Software libre. Las

principales metas de su diseño son: velocidad, simplicidad, multiplataforma y facilidad del desarrollo distribuido. Todo el código fuente de Apache está escrito en C, con un total aproximado de 185.000 líneas de código.

El servidor HTTP Apache es un servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etcétera), Windows y otras, que implementa el protocolo HTTP/4.3 y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 4.3, pero más tarde fue reescrito por completo. Su nombre se debe a que originalmente Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. Era, en inglés, a patchy server (un servidor "parcheado").

El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation.

Apache presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración. Apache tiene amplia aceptación en la red: en el 2005, Apache fue el servidor HTTP más usado, siendo el servidor HTTP del 70% de los sitios Web en el mundo y creciendo aún su cuota de mercado (estadísticas históricas y de uso diario proporcionadas por Netcraft).

El núcleo 2.x de Apache tiene varias mejoras clave sobre el núcleo de Apache 1.x. Estas mejoras incluyen threads de UNIX, mejor soporte para plataformas no Unix (como Windows), un nuevo API, y soporte de IPv6.

2.8.1 Módulos

La arquitectura del servidor Apache es muy modular. El servidor consta de un sección core y mucha de la funcionalidad que podría considerarse básica para un servidor Web es provista por módulos.

2.8.2 Uso

Apache es usado principalmente para enviar páginas web estáticas y dinámicas en la World Wide Web. Muchas aplicaciones web están diseñadas asumiendo como ambiente de implantación a Apache, o que utilizarán características propias de este servidor web.

Apache es el componente de servidor web en la popular plataforma de aplicaciones LAMP, junto a MySQL y los lenguajes de programación PHP/Perl/Python (y ahora también Ruby).

Este servidor web es redistribuido como parte de varios paquetes propietarios de software, incluyendo la base de datos Oracle y el IBM WebSphere application

server. Mac OS X integra apache como parte de su propio servidor web y como soporte de su servidor de aplicaciones WebObjects. Es soportado de alguna manera por Borland en las herramientas de desarrollo Kylix y Delphi. Apache es incluido con Novell NetWare 6.5, donde es el servidor web por defecto, y en muchas distribuciones Linux.

Apache es usado para muchas otras tareas donde el contenido necesita ser puesto a disposición en una forma segura y confiable. Un ejemplo es al momento de compartir archivos desde una computadora personal hacia Internet. Un usuario que tiene Apache instalado en su escritorio puede colocar arbitrariamente archivos en la raíz de documentos de Apache, desde donde pueden ser compartidos.

Los programadores de aplicaciones web a veces utilizan una versión local de Apache con el fin de pre visualizar y probar código mientras éste es desarrollado.

Microsoft Internet Information Services (IIS) es el principal competidor de Apache, así como Sun Java System Web Server de Sun Microsystems y un anfitrión de otras aplicaciones como Zeus Web Server. Algunos de los más grandes sitios web del mundo están ejecutándose sobre Apache. La capa frontal (front end) del motor de búsqueda Google está basada en una versión modificada de Apache, denominada Google Web Server (GWS). Muchos proyectos de Wikimedia también se ejecutan sobre servidores web Apache.

2.8.3 Configuración

La mayor parte de la configuración se realiza en el fichero `apache2.conf` o `httpd.conf`, según el sistema donde esté corriendo. Cualquier cambio en este archivo requiere reiniciar el servidor, o forzar la lectura de los archivos de configuración nuevamente.

2.8.4 Licencia

La licencia de software bajo la cual el software de la fundación Apache es distribuido es una parte distintiva de la historia de Apache HTTP Server y de la comunidad de código abierto. La Licencia Apache permite la distribución de derivados de código abierto y cerrado a partir de su código fuente original.

La Free Software Foundation no considera a la Licencia Apache como compatible con la versión 2 de la GNU General Public License (GPL), en la cual el software licenciado bajo la Apache License no puede ser integrado con software distribuido bajo la GPL:

Este es software libre pero es incompatible con la GPL. La Apache Software License es incompatible con la GPL porque tiene un requerimiento específico que no está incluido en la GPL: tiene ciertos casos de terminación de patentes que la GPL no requiere. No consideramos que dichos casos de terminación de patentes son inherentemente una mala idea, pero a pesar de ello son incompatibles con la GNU GPL.

Sin embargo, la versión 3 de la GPL incluye una provisión (Sección 7e) que le permite ser compatible con licencias que tienen cláusulas de represalia de patentes, incluyendo a la Licencia Apache.

El nombre Apache es una marca registrada y puede ser sólo utilizada con el permiso expreso del dueño de la marca.

2.9 INSTALACIÓN DE APACHE

2.9.1 Descargar

Puede descargar Apache desde la sección de descargas del sitio Web de Apache el cual tiene varios mirrors. Para la mayoría de los usuarios de Apache que tienen sistemas tipo Unix. El proceso de compilación (descrito más abajo) es fácil, y permite adaptar el servidor Apache a sus necesidades. Además, las versiones disponibles en archivos binarios no están siempre actualizadas con las últimas modificaciones en el código fuente. Si se descarga un binario, siga las instrucciones contenidas en el archivo `INSTALL.bindist` incluido en la distribución

Después de la descarga, es importante que verifique que el archivo descargado del servidor HTTP Apache está completo y sin modificaciones. Esto puede hacerlo comparando el archivo descargado (.tgz) con su firma PGP. Instrucciones detalladas de cómo hacer esto están disponibles en la sección de descargas junto con un ejemplo de cómo usar PGP.

2.9.2 Inicio de Apache

Si el puerto especificado en la directiva Listen del fichero de configuración es el que viene por defecto, es decir, el puerto 80 (o cualquier otro puerto por debajo del 1024), entonces es necesario tener privilegios de usuario root (superusuario) para iniciar Apache, de modo que pueda establecerse una conexión a través de esos puertos privilegiados. Una vez que el servidor Apache se ha iniciado y ha completado algunas tareas preliminares, tales como abrir sus ficheros log, lanzarán varios procesos, procesos hijo, que hacen el trabajo de escuchar y atender las peticiones de los clientes. El proceso principal, httpd continúa ejecutándose como, pero los procesos hijo se ejecutan con menores privilegios de usuario. Esto lo controla el Módulo de MultiProcesamiento (MPM) seleccionado.

Si todo va bien durante el arranque, la sesión de Terminal se suspenderá un momento y volverá a estar activa casi inmediatamente. Esto quiere decir que el servidor está activo y funcionando. Puede usar su navegador para conectarse al servidor y ver la página de prueba que hay en el directorio DocumentRoot y la copia local de esta documentación a la que se puede acceder desde esa página.

2.10 INTERNET INFORMATION SERVICES (IIS)



Figura II.6. Logotipo Internet Information Services

Internet Information Services o IIS es un servidor web y un conjunto de servicios para el sistema operativo Microsoft Windows. Originalmente era parte del Option Pack para Windows NT. Luego fue integrado en otros sistemas operativos de Microsoft destinados a ofrecer servicios, como Windows 2000 o Windows Server 2003. Windows XP Profesional incluye una versión limitada de IIS. Los servicios que ofrece son: FTP, SMTP, NNTP y HTTP/HTTPS.

Este servicio convierte a una PC en un servidor web para Internet o una intranet, es decir que en las computadoras que tienen este servicio instalado se pueden publicar páginas web tanto local como remotamente.

Los servicios de Internet Information Services proporcionan las herramientas y funciones necesarias para administrar de forma sencilla un servidor web seguro.

El servidor web se basa en varios módulos que le dan capacidad para procesar distintos tipos de páginas. Por ejemplo, Microsoft incluye los de Active Server

Pages (ASP) y ASP.NET. También pueden ser incluidos los de otros fabricantes, como PHP o Perl.

2.10.1 Historia

IIS fue inicialmente lanzado como un conjunto de servicios basados en Internet para Windows NT 3.51. IIS 2.0 siguió agregando soporte para el sistema operativo Windows NT 4.0 y IIS 3.0 introdujo las Active Server Pages, una tecnología de scripting dinámico.

IIS 4.0 eliminó el soporte para el protocolo Gopher y fue puesto con Windows NT como un CD-ROM de "Paquete Opcional" separado.

La versión actual de IIS es la 7.5 para Windows Server 2008 y IIS 5.1 para Windows XP Professional. IIS 5.1 para Windows XP es una versión compacta del IIS que soporta sólo 10 conexiones simultáneas y sólo un sitio web. IIS 6.0 ha agregado soporte para IPv6.

Windows Vista viene con IIS 7.0 preinstalado. No limitará el número de conexiones permitidas pero limitará el flujo de tareas basándose en las solicitudes activas concurrentes, mejorando el uso y el rendimiento en escenarios punto-a-punto (peer-to-peer).

2.10.2 Características

La arquitectura de IIS 7 es modular. Módulos, extensiones también se llama, se pueden añadir o eliminar en forma individual, para que sólo los módulos necesarios para la funcionalidad específica tiene que ser instalado. IIS 7 incluye módulos nativos como parte de la instalación completa. Estos módulos son elementos individuales que utiliza el servidor para procesar las solicitudes y son los siguientes:

HTTP módulos - Se utiliza para realizar tareas específicas de HTTP en la canalización de solicitudes de procesamiento, tales como la respuesta a la información y consultas enviadas en las cabeceras de los clientes, devolver los errores HTTP, y el desvío de las peticiones.

Los módulos de seguridad - Se utiliza para realizar tareas relacionadas con la seguridad en la canalización de solicitudes de procesamiento, como la especificación de esquemas de autenticación, la realización de la autorización de URL y filtrado de las solicitudes.

Los módulos de contenido - Se utiliza para realizar tareas relacionadas con el contenido de la canalización de solicitudes de procesamiento, tales como el procesamiento de solicitudes de archivos estáticos, devolviendo una página por defecto cuando un cliente no especifica un recurso en una solicitud, y listar el contenido de un directorio.

Los módulos de compresión - Se utiliza para realizar tareas relacionadas con la compresión en la canalización de solicitudes de procesamiento, tales como la compresión de las respuestas, la aplicación de la transferencia de compresión Gzip codificación de las respuestas, y la realización de pre-compresión de contenido estático.

Módulos de almacenamiento en caché - Se utiliza para realizar tareas relacionadas con el almacenamiento en caché en la canalización de solicitudes de procesamiento, tales como el almacenamiento de la información procesada en la memoria en el servidor y el uso de contenido almacenado en caché en las solicitudes posteriores de los mismos recursos.

Inicio de sesión y los módulos de diagnóstico - Se utiliza para realizar tareas relacionadas con el registro y diagnóstico de la canalización de solicitudes de procesamiento, tales como la transmisión de información y el estado de procesamiento de HTTP.sys para el registro, reporte de eventos, y el seguimiento de las solicitudes que se está ejecutando en los procesos de trabajo.

IIS 6.0 y un mayor apoyo a los mecanismos de autenticación siguientes:

- La autenticación anónima
- Autenticación de acceso básica
- Recopilación de autenticación de acceso
- Autenticación de Windows integrada
- UNC autenticación

- .NET Passport (eliminado en Windows Server 2008 e IIS 7.0)
- Certificado de autenticación

IIS 7.5 incluye las siguientes características de seguridad adicional o mejorada:

- Asignación de certificados de cliente
- Seguridad IP
- Filtro de solicitudes
- Autorización de direcciones URL

2.10.3 IIS expreso

IIS Express, una versión ligera de IIS, está disponible como un servidor independiente del freeware y puede ser instalado en Windows XP con Service Pack 3 y versiones posteriores de Microsoft Windows. IIS 7.5 Express es compatible con sólo los protocolos HTTP y HTTPS. IIS Express se puede descargar por separado o como parte de Microsoft WebMatrix.

2.10.4 Extensiones

IIS libera nuevos módulos cuentan con una versión a la versión principal de añadir nuevas funcionalidades. Las siguientes extensiones están disponibles para IIS 7.5:

Servicio de publicación FTP - Permite creadores de contenidos web publicar contenido de forma segura a los servidores web IIS 7 con la autenticación basada en SSL y la transferencia de datos.

Paquete de Administración - Añade apoyo a la administración de interfaz de usuario para las funciones de administración en IIS 7, incluyendo la autorización de ASP.NET, los errores de medida, configuración FastCGI, y el filtrado de solicitudes.

Solicitud de aplicación de enrutamiento - Proporciona un módulo de enrutamiento basado en proxy que redirecciona las peticiones HTTP a los servidores de contenido basado en los encabezados HTTP, variables de servidor, y los algoritmos de balanceo de carga.

Administrador de Base de datos - Permite una fácil gestión de bases de datos local y remota desde el Administrador de IIS.

Servicios de Medios de Comunicación - Se integra una plataforma de entrega de medios con IIS para gestionar y administrar la entrega de medios ricos y otros contenidos web.

Módulo de reescritura de URL - Proporciona un mecanismo basado en reglas de reescritura para cambiar las URL de solicitud antes de que sean procesados por el servidor web.

WebDAV - permite a los autores publicar contenido Web de forma segura a IIS 7 servidores web, y permite a los administradores web y hosters administrar la configuración de WebDAV con IIS 7 y herramientas de gestión de configuración.

Herramienta de implementación web - Sincroniza IIS 6.0 y los servidores IIS 7, migra un servidor IIS 6.0 a IIS 7, y despliega aplicaciones web a un servidor IIS 7.

2.10.5 Versiones

- IIS 1.0, Windows NT 3.51 Service Pack 3
- IIS 2.0, Windows NT 4.0
- IIS 3.0, Windows NT 4.0 Service Pack 3
- IIS 4.0, Windows NT 4.0 Option Pack
- IIS 5.0, Windows 2000
- IIS 5.1, Windows XP Professional
- IIS 6.0, Windows Server 2003 y Windows XP Profesional x64 Edition
- IIS 7.0, Windows Vista (Solo Business y Ultimate) y Windows Server 2008
- IIS 7.5, Windows 7 y Windows Server 2008 R2

2.11 WINDOWS SERVER 2008



Figura II.7.Logotipo Windows Server 2008

Microsoft Windows Server 2008 está diseñado para ofrecer a las organizaciones la plataforma más productiva para virtualización de cargas de trabajo, creación de aplicaciones eficaces y protección de redes. Ofrece una plataforma segura y de fácil administración, para el desarrollo y alojamiento confiable de aplicaciones y servicios web. Del grupo de trabajo al centro de datos, Windows Server 2008 incluye nuevas funciones de gran valor y eficacia y mejoras impactantes en el sistema operativo base.

2.11.1 Características

2.11.1.1 Más control

Windows Server 2008 proporciona a los profesionales de TI más control sobre sus servidores e infraestructura de red y les permite centrarse en las necesidades críticas del negocio. Capacidades mejoradas en secuencias de comandos y automatización de tareas, como las que ofrece Windows PowerShell, ayudan a los profesionales de TI a automatizar tareas comunes de TI. La instalación y administración basadas en funciones con Administrador del Servidor facilita la tarea de administrar y proteger las múltiples funciones de servidor en una empresa. La nueva consola del Administrador del servidor proporciona un único origen para administrar la configuración del servidor y la información del sistema.

El personal de TI puede instalar sólo las funciones y características que sean necesarias, y hay asistentes que automatizan muchas de las tareas de implementación de sistemas que tardan más tiempo. Herramientas mejoradas de administración del sistema, como el Monitor de rendimiento y confiabilidad, ofrecen información sobre sistemas y alertan al personal de TI sobre problemas potenciales antes de que sucedan.

2.11.1.2 Mayor protección

Windows Server 2008 proporciona una serie de tecnologías de seguridad nuevas y mejoradas, que aumentan la protección del sistema operativo al ofrecer una base sólida para la dirigir y construir un negocio. Incluye innovaciones de seguridad, como PatchGuard, que reducen la exposición a **Particiones fuertes**: una máquina virtual (VM) funciona como un contenedor independiente de sistema operativo, completamente aislado de otras máquinas virtuales que se ejecutan en el mismo servidor físico.

Seguridad para el hardware: características como prevención de ejecución de datos (DEP) se encuentran disponibles en el hardware más reciente para servidores y ayudan a evitar la ejecución de los virus y los gusanos más predominantes.

Windows Server virtualization: WSv ayuda a evitar la exposición de las VM que contienen información confidencial y protege también al sistema operativo host subyacente del riesgo que comporta un sistema operativo invitado.

Características de seguridad de red: permite la traducción de direcciones de red (NAT) automática, firewall y protección de acceso a redes (NAP).

Base de equipos de confianza mínima: ofrece una superficie de ataque reducida y una arquitectura de virtualización simplificada y ligera. Esta característica mejora la confiabilidad de equipos virtuales basados en WSv.

Ataques del núcleo, lo que produce un entorno de servidor más seguro y estable. El sistema de protección de servicios de Windows ayuda a mantener más seguros los sistemas al evitar que los servicios críticos de servidor estén en riesgo por actividades anormales en el sistema de archivos, registro, o red. La seguridad también se mejora en el sistema operativo Windows Server 2008 por medio de protección de acceso a redes (NAP), controlador de dominio de sólo lectura (RODC), mejoras en la infraestructura de clave pública (PKI), un nuevo firewall de Windows bidireccional y compatibilidad con criptografía de última generación.

2.11.1.3 Mayor flexibilidad

Windows Server 2008 está diseñado para permitir que los administradores modifiquen su infraestructura para adaptarla a las necesidades cambiantes del negocio y continuar siendo ágiles. Se mejora la flexibilidad para trabajadores móviles mediante tecnologías que permiten que los usuarios ejecuten programas desde cualquier ubicación remota, como RemoteApp y Terminal Services Gateway. Windows Server 2008 acelera la implementación y el mantenimiento de sistemas de TI con Servicios de Implementación de Windows (WDS) y ayuda en la consolidación de servidores con Windows Server virtualization (WSv). Para

organizaciones que necesitan controladores de dominio en sucursales, Windows Server 2008 ofrece una nueva opción de configuración: el Controlador de Dominio de sólo lectura (RODC), que evita exponer las cuentas si el Controlador de Dominio estuviera en riesgo.

2.11.2 VISUALIZACIÓN

2.11.2.1 Seguridad

La seguridad es un desafío fundamental en cada implementación de servidor. Un servidor que aloja múltiples máquinas virtuales (VM), también conocidos como servidores consolidados, está expuesto a los mismos riesgos de seguridad que los servidores no consolidados, pero agrega el desafío de separación de funciones de administrador. WSv ayuda a aumentar la seguridad de servidores consolidados y resuelve el desafío de separación de funciones de administrador. WSv lo lleva a cabo por medio de las siguientes características:

La configuración de un servidor consolidado que ofrezca los mejores entornos de seguridad y sistema operativo para cada aplicación puede presentar en ciertas ocasiones un desafío difícil. Debido a que WSv crea un entorno donde es posible configurar cada carga de trabajo con un entorno de sistema operativo y perfil de seguridad ideales, WSv resuelve el desafío de la separación de funciones en un

servidor consolidado. WSv protege a las VM del sistema operativo host y viceversa, al permitir que las VM se ejecuten en una cuenta de servicio sólo con los privilegios necesarios. Con WSv, el sistema operativo host está protegido y una VM en riesgo está limitada en el daño que podría causar a otras VM.

2.11.2.2 Fuerte aislamiento

La virtualización del servidor permite que coexistan cargas de trabajo con requisitos de recursos diferentes en el mismo servidor host. WSv ofrece varias características que facilitan el uso eficaz de los recursos físicos del servidor host:

- **Asignación flexible de memoria:** se puede asignar una cantidad máxima y una cantidad mínima de memoria RAM garantizada a las máquinas virtuales. Esta característica permite que los administradores creen una configuración de WSv que equilibre las necesidades individuales del recurso de VM frente al rendimiento total del servidor de WSv.
- **Adición dinámica de hardware:** WSv puede agregar dinámicamente procesadores lógicos, memoria, adaptadores de red y almacenamiento a sistemas operativos invitados compatibles, mientras se encuentran en ejecución. Esta característica facilita la asignación granular de capacidades de procesamiento host de WSv a los sistemas operativos invitados.
- **Configuración flexible de red:** WSv ofrece características avanzadas de red para las VM, que incluyen NAT, firewall y asignación de VLAN. Esta

flexibilidad se puede usar para crear una configuración de WSv más compatible con los requisitos de seguridad de red.

Las características de asignación flexible de memoria, adición dinámica de hardware y configuración flexible de red de WSv facilitan una respuesta más eficaz a las cargas dinámicas de servidor. Por ejemplo, la carga de trabajo de procesamiento de fin de período es con frecuencia varias veces mayor que el promedio en algunas aplicaciones de línea de negocios (LOB). WSv se puede usar con sistemas operativos invitados compatibles, para asignar dinámicamente recursos adicionales de memoria y procesador a una VM en ejecución y administrar los requisitos ampliados de procesamiento sin reiniciar el sistema operativo invitado. Con suficientes recursos de servidor host, este cambio no disminuye el rendimiento de las demás VM que se ejecutan en el host.

2.11.2.3 Rendimiento

Los avances e integración del diseño con hardware que reconoce la virtualización permite a WSv virtualizar cargas de trabajo mucho más exigentes que en versiones anteriores y con mayor flexibilidad en la asignación de recursos.

Los avances de rendimiento incluyen:

- **Arquitectura de virtualización ligera, de baja sobrecarga, basada en Hypervisor de 64 bits:** hardware preparado para virtualización (Intel VT y

la tecnología "Pacífica" de AMD) que permite rendimientos mayores del sistema operativo invitado.

- **Compatibilidad con múltiples núcleos.** A cada VM se le pueden asignar hasta ocho procesadores lógicos: esto permite la virtualización de grandes cargas de trabajo, con cálculo intensivo, que aprovechan los beneficios del procesamiento en paralelo de núcleos de VM con procesadores múltiples.
- **Compatibilidad de sistemas operativos host e invitado de 64 bits:** WSv se ejecuta en la versión de 64 bits de Windows Server 2008 para ofrecer acceso a grandes grupos de memoria para las VM invitadas. Cargas de trabajo intensivas en memoria que se verían muy afectadas por extensas paginaciones si se ejecutaran en sistemas operativos de 32 bits, se pueden virtualizar correctamente en WSv. WSv también es compatible con sistemas operativos invitados de 64 y 32 bits que se ejecutan en el mismo servidor consolidado.

Compatibilidad con Server Core. WSv puede usar una instalación Server Core de Windows Server 2008 como sistema operativo host. La superficie mínima de instalación y baja sobrecarga de Server Core dedica la mayor cantidad posible de la capacidad de procesamiento de servidor host a las VM en ejecución.

Acceso a disco de paso. Los sistemas operativos invitados se pueden configurar para tener acceso de forma directa a almacenamiento local o de red de área de almacenamiento (SAN) iSCSI, lo que ofrece mayores rendimientos en aplicaciones con muchas operaciones de E/S, como SQL Server o Microsoft Exchange.

Muchas cargas de trabajo de servidor demandan mucho procesamiento de servidor y subsistemas de E/S. Cargas de trabajo como SQL Server y Microsoft Exchange normalmente consumen mucha memoria y lastran el rendimiento de disco, y ha habido resistencia a virtualizar estas cargas de trabajo. El Hypervisor de 64 bits en WSv junto con características como el acceso a disco de paso hace posible y con frecuencia deseable virtualizar grandes cargas de trabajo.

2.11.2.4 Administración simplificada

En las instalaciones de centros de datos y sucursales remotas donde es posible implementar WSv, se necesitan fuertes capacidades de administración y automatización para ser totalmente conscientes del potencial de reducción de costos de la virtualización. WSv satisface este desafío con las siguientes capacidades de administración y automatización:

Administración extensible: WSv está diseñado para funcionar con Microsoft System Center Operations Manager (SCOM) y System Center Virtual Machine Manager (SCVMM). Estas herramientas de administración ofrecen informes, automatización, implementación y herramientas autoservicio de usuario para WSv.

Interfaz MMC 3.0 para administración de VM: la familiar interfaz Microsoft Management Console (MMC) se usa para administrar la configuración de WSv y los valores de VM, reduciendo sensiblemente el proceso de aprendizaje de WSv.

Interfaz del instrumental de administración de Windows (WMI): WSv incorpora un proveedor WMI que ofrece acceso a información de sistema y administración mediante secuencias de comandos.

Secuencias de comandos de PowerShell: la configuración de host y VM de WSv se configura a través de Windows PowerShell.

Administración de objetos de directivas de grupo (GPO): WSv usa las capacidades de administración de configuración de GPO para administrar la virtualización del host WSv y la configuración del equipo virtual.

Las capacidades de administración de SCOM y SCVMM hacen posible administrar de forma eficaz tanto instalaciones de centros de datos como instalaciones sumamente distribuidas de WSv. Por ejemplo, el acceso de secuencias de comandos al proveedor WMI en WSv se puede usar para automatizar las ventanas de mantenimiento en varios servidores host de WSv, al apagar VM invitadas, encenderlas en un servidor en espera, realizar el mantenimiento en el servidor host y, a continuación, restaurar las VM a su host original. Con la adición de System Center Virtual Machine Manager, esta operación se puede automatizar y realizar sin ningún tiempo de inactividad perceptible para muchas aplicaciones.

2.12 DEFINICIÓN DE MÁQUINA VIRTUAL

En informática una máquina virtual es un software que emula a una computadora y puede ejecutar programas como si fuese una computadora real. Este software en un principio fue definido como "un duplicado eficiente y aislado de una máquina física". La acepción del término actualmente incluye a máquinas virtuales que no tienen ninguna equivalencia directa con ningún hardware real.

Una característica esencial de las máquinas virtuales es que los procesos que ejecutan están limitados por los recursos y abstracciones proporcionados por ellas. Estos procesos no pueden escaparse de esta "computadora virtual".

Uno de los usos domésticos más extendidos de las máquinas virtuales es ejecutar sistemas operativos para "probarlos". De esta forma podemos ejecutar un sistema operativo que queramos probar (GNU/Linux, por ejemplo) desde nuestro sistema operativo habitual (Mac OS X por ejemplo) sin necesidad de instalarlo directamente en nuestra computadora y sin miedo a que se desconfigure el sistema operativo primario.

2.12.1 Tipos de máquinas virtuales

Las máquinas virtuales se pueden clasificar en dos grandes categorías según su funcionalidad y su grado de equivalencia a una verdadera máquina.

- Máquinas virtuales de sistema (en inglés System Virtual Machine)

- Máquinas virtuales de proceso (en inglés Process Virtual Machine)

2.12.2 Máquinas virtuales de sistema

Las máquinas virtuales de sistema, también llamadas máquinas virtuales de hardware, permiten a la máquina física subyacente multiplicarse entre varias máquinas virtuales, cada una ejecutando su propio sistema operativo. A la capa de software que permite la virtualización se la llama monitor de máquina virtual o "hypervisor". Un monitor de máquina virtual puede ejecutarse o bien directamente sobre el hardware o bien sobre un sistema operativo ("host operating system").

2.12.3 Aplicaciones de las máquinas virtuales de sistema

Varios sistemas operativos distintos pueden coexistir sobre la misma computadora, en sólido aislamiento el uno del otro, por ejemplo para probar un sistema operativo nuevo sin necesidad de instalarlo directamente.

La máquina virtual puede proporcionar una arquitectura de instrucciones (ISA) que sea algo distinta de la verdadera máquina. Es decir, podemos simular hardware.

Varias máquinas virtuales (cada una con su propio sistema operativo llamado sistema operativo "invitado" o "guest"), pueden ser utilizadas para consolidar servidores. Esto permite que servicios que normalmente se tengan que ejecutar en

computadoras distintas para evitar interferencias, se puedan ejecutar en la misma máquina de manera completamente aislada y compartiendo los recursos de una única computadora. La consolidación de servidores a menudo contribuye a reducir el coste total de las instalaciones necesarias para mantener los servicios, dado que permiten ahorrar en hardware.

La virtualización es una excelente opción hoy día, ya que las máquinas actuales (Laptops, desktops, servidores) en la mayoría de los casos están siendo "sub-utilizados" (gran capacidad de disco duro, memoria RAM, etc.), llegando a un uso de entre 30% a 60% de su capacidad. Al virtualizar, la necesidad de nuevas máquinas en una ya existente permite un ahorro considerable de los costos asociados (energía, mantenimiento, espacio, etc).

2.12.4 Máquinas virtuales de proceso

Una máquina virtual de proceso, a veces llamada "máquina virtual de aplicación", se ejecuta como un proceso normal dentro de un sistema operativo y soporta un solo proceso. La máquina se inicia automáticamente cuando se lanza el proceso que se desea ejecutar y se detiene para cuando éste finaliza. Su objetivo es el de proporcionar un entorno de ejecución independiente de la plataforma de hardware y del sistema operativo, que oculte los detalles de la plataforma subyacente y permita que un programa se ejecute siempre de la misma forma sobre cualquier plataforma.

El ejemplo más conocido actualmente de este tipo de máquina virtual es la máquina virtual de Java. Otra máquina virtual muy conocida es la del entorno .Net de Microsoft que se llama "Common Language Runtime".

2.12.5 Inconvenientes de las máquinas virtuales

Uno de los inconvenientes de las máquinas virtuales es que agregan gran complejidad al sistema en tiempo de ejecución. Esto tiene como efecto la ralentización del sistema, es decir, el programa no alcanzará la misma velocidad de ejecución que si se instalase directamente en el sistema operativo "anfitrión" (host) o directamente sobre la plataforma de hardware. Sin embargo, a menudo la flexibilidad que ofrecen compensa esta pérdida de eficiencia.

2.12.6 Técnicas

2.12.6.1 Emulación del hardware subyacente (ejecución nativa)

Esta técnica se suele llamar virtualización completa (full virtualization) del hardware, y se puede implementar usando un hypervisor de Tipo 1 o de Tipo 2:

- El tipo 1 se ejecuta directamente sobre el hardware
- El tipo 2 se ejecuta sobre otro sistema operativo.

Cada máquina virtual puede ejecutar cualquier sistema operativo soportado por el hardware subyacente. Así los usuarios pueden ejecutar dos o más sistemas operativos distintos simultáneamente en computadoras "privadas" virtuales.

El sistema pionero que utilizó este concepto fue la CP-40, la primera versión (1967) de la CP/CMS de IBM (1967-1972) y el precursor de la familia VM de IBM (de 1972 en adelante). Con la arquitectura VM, la mayor parte de usuarios controlan un sistema operativo monousuario relativamente simple llamado CMS que se ejecuta en la máquina virtual VM.

Actualmente tanto Intel como AMD han introducido prestaciones a sus procesadores x86 para permitir la virtualización de hardware.

2.12.6.2 Emulación de un sistema no nativo

Las máquinas virtuales también pueden actuar como emuladores de hardware, permitiendo que aplicaciones y sistemas operativos concebidos para otras arquitecturas de procesador se puedan ejecutar sobre un hardware que en teoría no soportan.

Algunas máquinas virtuales emulan hardware que sólo existe como una especificación. Por ejemplo:

- La máquina virtual P-Code que permitía a los programadores de Pascal crear aplicaciones que se ejecutasen sobre cualquier computadora con esta máquina virtual correctamente instalada.
- La máquina virtual de Java.
- La máquina virtual del entorno .NET.
- Open Firmware

Esta técnica permite que cualquier computadora pueda ejecutar software escrito para la máquina virtual. Sólo la máquina virtual en sí misma debe ser portada a cada una de las plataformas de hardware.

2.12.6.3 Virtualización a nivel de sistema operativo

Esta técnica consiste en dividir una computadora en varios compartimentos independientes de manera que en cada compartimento podamos instalar un servidor. A estos compartimentos se los llama "entornos virtuales". Desde el punto de vista del usuario, el sistema en su conjunto actúa como si realmente existiesen varios servidores ejecutándose en varias máquinas distintas. Dos ejemplos son las zonas de Solaris (Solaris Zones) y la técnica de Micro Partioning de AIX.

2.13 MEDIR EL RENDIMIENTO DESDE UN CLIENTE WINDOWS

2.13.1 Colasoft Capsa

Colasoft Capsa es el más poderoso pero sencillo monitor y analizador de red, diseñado para decodificación de paquetes y diagnóstico de redes. Con las características de monitoreo y análisis de datos en tiempo real, se puede capturar y decodificar el tráfico de la red transmitido sobre el sistema local y la red general.

Colasoft Capsa es la herramienta perfecta para administrar la red y sistemas, ya que permite un control detallado y exhaustivo de la red.

Entre las características más importantes están:

- Monitoreo en tiempo real
- Análisis y captura de paquetes, con gráficas por protocolo
- Decodificación de paquetes
- Interfaz de Usuario amigable.
- Módulo de análisis de correo electrónico (Se puede interferir contraseñas y contenido de todo el tráfico SMTP Y POP3)
- Módulo de análisis de accesos a la Web (Se puede ver todo el tráfico web por ip y usuario)
- Modulo de análisis de mensajería instantánea (Se puede ver todos los mensajes y sesiones de mensajería)
- Control del ancho de banda usado por usuario, protocolo, aplicación etc...

- Visualización flexible y clara.

2.13.2 WebServer Stress Tools

Webserver Stress Tool simula grandes números de usuarios accediendo un sitio web vía HTTP/HTTPS. El software puede simular hasta 10000 usuarios que independientemente van recorriendo su camino a través de una serie de URLs.

Permite patrones de URL simples, así como patrones complejos, a través de un archivo de Script. Basado en los parámetros que especifica, la aplicación no sólo pide el HTML de una URL, sino también frames, imágenes, archivos de Flash, etc., emulando el mismo comportamiento que un navegador web debería mostrar al acceder al sitio web.

Cada usuario es simulado con un hilo de ejecución separado con su propia información de sesión (ej. las cookies para cada usuario simulado se guardan separadamente) y "navega" las URLs independientemente de los otros usuarios - tal como en un uso real.

Las URLs pueden ser parametrizadas para cada usuario y la secuencia de URLs puede variar. Las peticiones POST y GET están soportadas así como la autenticación Básica HTTP y otras muchas opciones. La nueva funcionalidad de scripting permite crear patrones de URL altamente complejos para aplicaciones web de gran escala.

Webserver Stress Tool asegura que los incidentes críticos en su sitio web se resuelvan antes que tiren abajo sus recursos web. Asegura que sus sitios web y aplicaciones reciben los recursos de servidor que necesitan para garantizar una experiencia de usuario de gran calidad, y que está aprovechando al máximo la inversión de su tecnología de servidor a través de consistentes análisis y tests en profundidad. Webserver Stress Tool es la solución más costo-efectiva del mercado de la simulación de rendimiento, carga y test de estrés para su servidor web.

Muchos sitios web hoy en día tienen una misión de un negocio serio para ganar dinero. Y si eso es a través de proporcionar contenido personalizado y los servicios propios, a través de oportunidades de publicidad o la venta de productos al por menor, estos sitios web de alto tráfico y aplicaciones deben estar en funcionamiento en todo momento. Porque si el rendimiento disminuye, incluso un poco, los usuarios web volubles probablemente elijan los servicios de otro competidor.

2.13.3 El mensaje a los propietarios de sitios web es clara: probar y monitorear su sitio web.

Pocos sitios web, en su caso, llevan a cabo pruebas exhaustivas. Por lo general, se centró exclusivamente en la captura de bugs, muchos sitios web ignoran las pruebas de funcionalidad, pruebas de usabilidad y los elementos de pruebas de rendimiento y tres críticos en la definición de la experiencia del usuario con un sitio

web o aplicación web. En resumen, los webmasters y desarrolladores no sólo deben probar los errores, la prueba de si el sitio web hace lo que se pretende hacer (pruebas de funcionalidad) y comprobar si el usuario es capaz de realizar fácilmente las tareas y objetivos en la página web (pruebas de usabilidad), pero también debe comprobar si el usuario recibe los resultados de la página web en un tiempo aceptable (pruebas de rendimiento).

Webserver Stress Tool permite rápidamente determinar e identificar problemas de rendimiento, para que rápidamente se puedan corregir para evitar la insatisfacción de los usuarios y la pérdida potencial de ingresos.

A través de una interfaz intuitiva, parámetros flexibles de pruebas y elaboración de informes completos, Webserver Stress Tool proporciona la herramienta para incluir las pruebas de rendimiento como una parte regular de mantenimiento de aplicaciones web y la web y la implementación.

Una vez que su servidor web se ha desplegado con las configuraciones correctas (en base a las pruebas de rendimiento), también puede considerar el despliegue de una aplicación de monitoreo 24/7.

2.14 MEDIR EL RENDIMIENTO DESDE UN CLIENTE LINUX

2.14.1 Comando ab

Para darnos una idea de que capacidad de respuesta tiene nuestro servidor WEB podemos usar una utilidad que viene en el paquete apache-utils y que se llama ab.

AB es una herramienta de evaluación comparativa de Apache. Está diseñado para dar una impresión de cómo una instalación web funciona. Este en especial muestra cómo la instalación es capaz de servir muchas peticiones por segundo.

Es muy útil cuando estamos haciendo cambios en el servidor para aumentar la performance y en ese caso ab nos da una clara medida de como estos cambios afectan para bien o mal a nuestro servidor WEB.

Una vez instalado podemos usarlo para ejecutar un test de 100 conexión con un concurrencia de 10 conexiones a la vez podemos hacer

```
ab -n 100 -c 10 http://www.google.com/
```

La respuesta es como en la que se ve en la imagen

```

This is ApacheBench, Version 2.3 <$Revision: 655654 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking www.google.com (be patient).....done

Server Software:      gws
Server Hostname:      www.google.com
Server Port:          80

Document Path:        /
Document Length:      222 bytes

Concurrency Level:    10
Time taken for tests:  2.769 seconds
Complete requests:    100
Failed requests:      0
Write errors:         0
Non-2xx responses:   100
Total transferred:    81500 bytes
HTML transferred:     22200 bytes
Requests per second:  36.12 [#/sec] (mean)
Time per request:     276.876 [ms] (mean)
Time per request:     27.688 [ms] (mean, across all concurrent requests)
Transfer rate:        28.75 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median  max
Connect:     15   49  37.8      28   154
Processing:  162  215 105.1     181  1103
Waiting:     162  214 105.1     181  1103
Total:       185  264 115.4     214  1136

Percentage of the requests served within a certain time (ms)
 50%    214
 66%    292
 75%    326
 80%    338
 90%    351
 95%    360
 98%    612
 99%   1136
100%   1136 (longest request)

```

Figurall.8. Respuesta de ejecución del comando ab en un cliente Linux

Entre los parámetros de respuesta podemos ver algunos que son muy interesantes como por ejemplo la cantidad de pedidos que el servidor pudo servir por segundo (Requests per second), La tasa de transferencia (transfer rate), el tiempo que llevo hacer el test (Time Taken for test) y al final un detalle de porcentaje de conexiones según el tiempo que tomaron.

2.14.2 Comando `httping`

Se trata de una herramienta similar al ping, pero que en vez de enviar paquetes por ICMP a una IP, envía peticiones HTTP a un servidor web. De esta forma podemos medir la latencia del servicio. Lo que hace, es descargarse el HEAD de la web en cuestión y observar el tiempo de respuesta.

Es una herramienta para verificar el estado de los servidores web. Hay que enviar peticiones GET o HEAD al desolado servidor web y registrar ambas respuestas y el tiempo de viaje. El nombre de HTTPing proviene de la similitud sintáctica entre las opciones de la línea de comandos y del ping ICMP.

Las principales opciones (`httping --help`) que presenta `httping` son:

- **-g url**: URL a analizar (e.g. `-g www.google.es`)
- **-p port**: Puerto del servidor (e.g. `80`)
- **-c count**: Número de conexiones
- **-i interval**: Intervalo entre conexiones
- **-G**: Realiza un GET al recibir el HEAD
- **-b**: Muestra la velocidad de transferencia en KB/s (usada con `-G`)
- **-B**: Pide la página web comprimida.

- **-X**: Muestra el tráfico generado por la petición (usada con **-b**)

Ejemplos de ejecución

```

shell> httping -c 3 -Gbg http://www.server.com
PING www.server.com:80 (http://www.server.com):
connected to www.server.com:80, seq=0 time=805.42 ms 405KB/s
connected to www.server.com:80, seq=1 time=505.06 ms 573KB/s
connected to www.server.com:80, seq=2 time=506.63 ms 471KB/s
--- http://www.server.com ping statistics ---
3 connects, 3 ok, 0.00% failed
round-trip min/avg/max = 505.1/605.7/805.4 ms
Transfer speed: min/avg/max = 405/483/573 KB

shell> httping -c 3 -GbXg http://www.server.com
PING www.server.com:80 (http://www.server.com):
connected to www.server.com:80, seq=0 time=476.82 ms 659KB/s 32KB
connected to www.server.com:80, seq=1 time=460.99 ms 712KB/s 32KB
connected to www.server.com:80, seq=2 time=497.99 ms 653KB/s 32KB
--- http://www.server.com ping statistics ---
3 connects, 3 ok, 0.00% failed
round-trip min/avg/max = 461.0/478.6/498.0 ms
Transfer speed: min/avg/max = 653/675/712 KB

shell> httping -c 3 -GbBXg http://www.server.com
PING www.server.com:80 (http://www.server.com):
connected to www.server.com:80, seq=0 time=931.29 ms 90243KB/s 7KB (compres
connected to www.server.com:80, seq=1 time=753.34 ms 10137KB/s 7KB (compres
connected to www.server.com:80, seq=2 time=696.43 ms 10051KB/s 7KB (compres
--- http://www.server.com ping statistics ---
3 connects, 3 ok, 0.00% failed
round-trip min/avg/max = 696.4/793.7/931.3 ms
Transfer speed: min/avg/max = 902436/973762/1013724 KB

```

Figura II.9. Ejemplo de ejecución comando httping

2.14.3 Comando httperf

Este proyecto de **HP** tiene más opciones para “atacar” a los servidores, ya que el concepto de sesión que incluye se parece más a lo que es un cliente típico bajo HTTP, donde puedes simular “ráfagas” de actividad para múltiples clientes conectados en forma concurrente.

El nivel de detalle de las estadísticas llega al punto de reportar el porcentaje de uso del CPU, para poder descartar corridas donde **httperf** haya tenidos que competir por CPU afectando el proceso de medición.

El httperf es un software para el funcionamiento de medición del web server. Proporciona una facilidad flexible para generar varias cargas de trabajo HTTP y para el funcionamiento de medición del servidor.

El foco del httperf está no en la ejecución de una prueba patrón particular sino en el abastecimiento de una herramienta robusta, de alto rendimiento que facilite la construcción de las pruebas patrones micro- y del macronivel.

Las tres características de distinción del httperf son su robustez, que incluye la capacidad de generar y de sostener la sobrecarga del servidor, ayuda para los protocolos de HTTP/1.1 y del SSL, y su extensibilidad a los nuevos generadores de la carga de trabajo y a las medidas de funcionamiento.

Httpperf es una herramienta para medir el rendimiento de servidores web. Habla el protocolo HTTP, tanto en su versión HTTP/1.0 HTTP/1.1 y ofrece una variedad de generadores de carga de trabajo. Mientras se ejecuta, realiza un seguimiento de un número de parámetros de rendimiento que se resumen en forma de estadísticas.

La operación más básica de httperf es generar un número fijo de peticiones HTTP GET y mide cuántas respuestas regresan desde el servidor y en qué proporción llegan las respuestas.

```
Total: connections 30000 requests 29997 replies 29997 test-duration 299.992 s
Connection rate: 100.0 conn/s (10.0 ms/conn, <=14 concurrent connections)
Connection time [ms]: min 1.4 avg 3.0 max 163.4 median 1.5 stddev 7.3
Connection time [ms]: connect 0.6
Connection length [replies/conn]: 1.000

Request rate: 100.0 req/s (10.0 ms/req)
Request size [B]: 75.0

Reply rate [replies/s]: min 98.8 avg 100.0 max 101.2 stddev 0.3 (60 samples)
Reply time [ms]: response 2.4 transfer 0.0
Reply size [B]: header 242.0 content 1010.0 footer 0.0 (total 1252.0)
Reply status: 1xx=0 2xx=29997 3xx=0 4xx=0 5xx=0

CPU time [s]: user 94.31 system 205.26 (user 31.4% system 68.4% total 99.9%)
Net I/O: 129.6 KB/s (1.1*10^6 bps)

Errors: total 3 client-timo 0 socket-timo 0 connrefused 3 connreset 0
Errors: fd-unavail 0 addrunavail 0 fltab-full 0 other 0
```

Figura II.10. Respuesta de un servidor web a 30000 peticiones con el comando httpperf

CAPÍTULO III

COMPARATIVA DE SERVIDORES

3.1 INTRODUCCION

Con el fin de llevar a cabo la medición y la comparación del rendimiento, se instala en las máquinas candidatas la aplicación HTML o pagina web diseñada usando la programación para ello.

Así aseguramos que las mismas características estén alojadas en ambos servidores y las acciones para medirlas no presenten diferencias en los resultados que se desea obtener.

La carga de trabajo se refiere a la capacidad máxima que tiene el servidor web (hardware y software), para atender un conjunto de usuarios de manera simultánea. Por ello, las actividades de esta etapa tienen relación con comprobar,

de manera anticipada, el funcionamiento que tendrá el servidor del Sitio Web cuando esté en plena operación.

Las pruebas en este caso consisten en simular una carga de trabajo similar y superior a la que tendrá cuando el sitio esté funcionando, con el fin de detectar si el software instalado (programas y aplicaciones) cumple con los requerimientos de muchos usuarios simultáneos y también si el hardware (servidor y el equipamiento computacional de redes y enlace que lo conecta a Internet) es capaz de soportar la cantidad de visitas esperadas.

Es importante considerar que si el servidor está en las dependencias de un tercero que entrega el servicio de alojamiento del Sitio Web (hosting), se le debe solicitar a dicho proveedor un informe en que dé a conocer las características de carga de la solución de hardware y software sobre la cual funciona el Sitio Web de la institución.

Hay diversos software en el mercado que está orientado a este tipo de simulaciones, todos los cuales ofrecen características similares. Entre los datos más relevantes que es posible obtener se cuenta:

- Tiempo de acceso de los usuarios a los datos
- Volumen de datos y ancho de banda utilizado
- Archivos solicitados y tiempos usados en transferencia de datos
- Tiempo de espera de los usuarios tras hacer un clic
- Tiempo de respuesta a clicks de usuarios
- Niveles de error existentes tras clicks de usuarios

3.2 PROCEDIMIENTO PARA REALIZAR LAS PRUEBAS

El procedimiento para la realización de la comparación de los Servidores Apache e IIS es definido en el siguiente flujo de pasos:

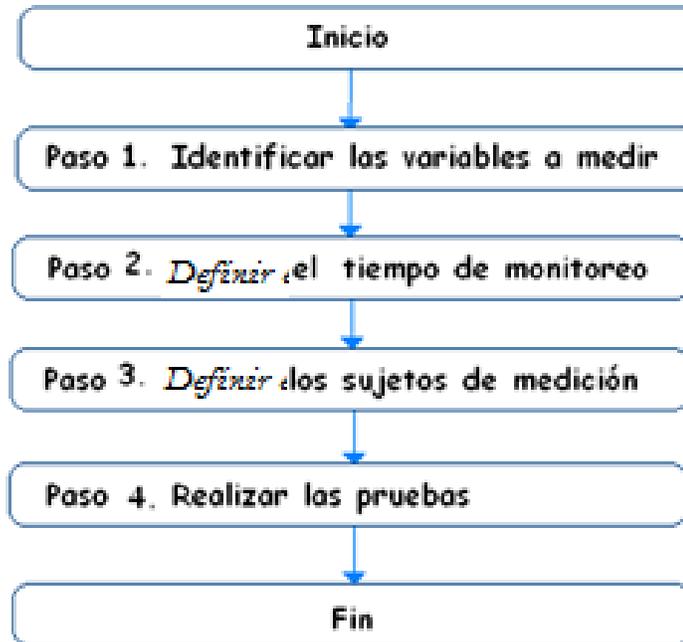


Figura II.11.Diagrama de Flujo para la elaboración de la Prueba de Rendimiento

3.2.1 Ambiente de Prueba

El ambiente de prueba para el desarrollo del estudio comparativo es el indicado en la Fig II.12.

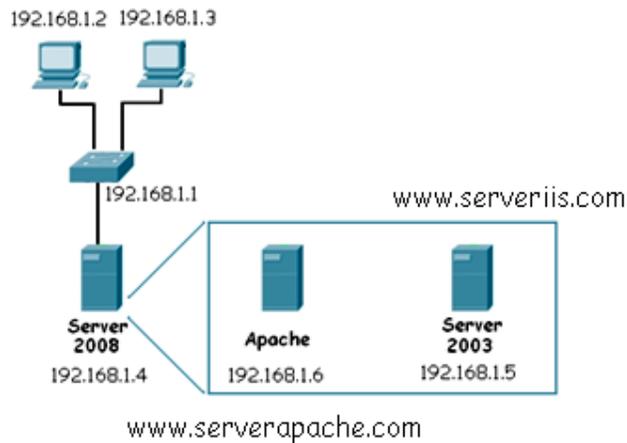


Figura II.12. Estructura de Red para realizar las pruebas de rendimiento

Las características del equipo que se utiliza como escenario para la comparativa de los servidores Web se describen en la Tabla III.4

3.2.3 Paso 1: Identificación de las variables

Las variables representan las características que suponemos, van a ser medidas o cuantificadas y que influyen en la comparación del rendimiento de los servidores. Puesto que, para la realización de las pruebas se le asignarán diferentes valores

Se han definido las siguientes variables para Análisis de Trafico, Análisis de Rendimiento, Análisis de Tiempos, las cuales se detallan en las siguientes tablas:

Tabla III.I.- Definición de variables a medir, Análisis de Tráfico

Variables	Concepto
<p>Análisis de Tráfico:</p> <p>Tráfico total</p> <p>Tráfico Broadcast</p> <p>Tráfico Multicast</p>	<p>Si en la red hay un tráfico excesivo de Broadcast o Multicast esto representa un problema sobre el rendimiento.</p>
<p>Análisis IP</p>	<p>Usado tanto por el cliente como por el servidor para la comunicación de datos, analizar cuál de los dos servidores genera más paquetes en un mismo ambiente de pruebas.</p>
<p>Análisis TCP:</p> <p>Paquetes SYN TCP</p> <p>Paquetes FIN TCP</p> <p>Conexiones Inicializadas</p> <p>Conexiones Establecidas</p>	<p>SYN: Paquetes que indican un pedido para establecer una conexión, se contabiliza el numero que se intercambian en la forma cliente-servidor</p> <p>FIN: Paquetes de interrupción o final de conexión, cuántos de los paquetes de intercambio tienen errores y provocan un final en la conexión</p>
<p>Análisis HTTP:</p> <p>Conexiones HTTP</p>	<p>Una conexión HTTP habilita a un paquete para obtener acceso a un servidor web mediante HTTP</p>

Peticiones HTTP	<p>para enviar o recibir archivos.</p> <p>Una petición HTTP indica al servidor que hacer con el URL, una petición habitual utiliza el método GET para pedirle al servidor que devuelva el URL solicitado.</p>
<p>Análisis DNS:</p> <p>Peticiones de DNS</p> <p>Respuestas de DNS</p> <p>Errores</p>	Realizar peticiones de resolución de nombres y esperar respuestas, analizar el numero de paquetes que se originan y si son correctos.
<p>Análisis NetBios</p> <p>Netbios TCP</p> <p>Netbios UDP</p>	Analizar los paquetes que se encarga de establecer la sesión y mantener las conexiones con el servidor
<p>Análisis ARP</p> <p>Peticiones</p> <p>Respuestas</p>	Interrogar a los equipos de la red para averiguar sus direcciones físicas, verificar cuantos paquetes se originan por parte de cada servidor
Tiempo de respuesta	Tiempo estimado de respuesta de los servidores al recibir peticiones

Elaborado por: Autora de Tesis

Tabla III.II.Definición de variables a medir, Análisis de Rendimiento

Variables	Concepto
Ancho de banda	Con el fin de monitorear la eficiencia de la red, se analiza tanto en el cliente como en el servidor
Análisis de Eficiencia Tráfico de Red Memoria del sistema disponible Carga del CPU Local	Cuanto tráfico de red en total se genera si se aumenta el número de usuarios, cuanta memoria tenemos disponible y a la vez cuanta carga de procesador produce ese aumento.

Elaborado por: Autora de Tesis

Tabla III.III.- Definición de variables a medir, Análisis de Tiempos

Variable	Concepto
Latencia con paquetes ICMP (ping)	Mediante el comando ping desde un cliente Windows analizamos la demora en la propagación y transmisión de paquetes
Latencia con paquete IP (ip)htping	Mediante el comando htting desde un cliente Linux analizamos la demora en

	la propagación y transmisión de paquetes
Tiempo de ida y Vuelta ab	Calcular el tiempo que tarda un paquete enviado desde un emisor en volver a este mismo emisor habiendo pasado por el receptor de destino.
Peticiones por segundo ab	Número de peticiones por segundo que el servidor soporta antes de presentar errores
Tiempo por petición ab	Tiempo promedio que transcurre para atender a las peticiones que se generan
Velocidad de Transferencia htping	Variable que se mide en KB para calcular
Uso del CPU durante las pruebas httpperf	Porcentaje de uso del CPU tanto del cliente como del servidor

Elaborado por: Autora de Tesis

3.2.4 Paso 2: Decidir los métodos y el tiempo de monitoreo

Los métodos más simples de monitoreo se basan específicamente en el tiempo que se demora un servidor en atender peticiones y el número de peticiones que puede soportar el servidor cuando es sometido a carga de usuarios.

Así, para las pruebas de rendimiento tomamos para cada nivel los siguientes métodos:

1. Método de carga o estrés de usuarios

- **Clics:** Con el cual se simula un número determinado de usuarios que están revisando la aplicación web al mismo tiempo y haciendo clic en los enlaces que tiene el diseño, es decir simular la visita de una página y su contenido completo, para ello, se realizarán 1000 clics por cada usuario de un total de 10.
- **Tiempo con carga constante:** prueba de funcionamiento con carga constante durante un tiempo determinado, para el caso que se menciona tomaremos 2 minutos de base para pruebas.
- **Tiempo con carga variable:** Prueba de funcionamiento con una carga que se va incrementando por un tiempo de 2 minutos.

2. Método de peticiones y tiempo de atención a las mismas

Tiempo: Se estima un tiempo de monitoreo inicial de 2 minutos durante el cual los servidores están sometidos a los métodos de carga mencionados y así medir los tiempos de respuestas.

3.2.5 Paso 4: Definir los sujetos de medición

Características técnicas del sujeto donde se realizarán las pruebas:

Tabla III.IV.Características Técnicas del sujeto sometido a prueba

Nombre	Características
SERVIDOR	<p>Procesador: Intel Pentium 4 Tehama I850</p> <p>RAM: 1GB</p> <p>Memoria en Disco:</p> <p>Tarjeta de Red:</p> <p>S.O principal: Windows Server 2008</p> <p>S.O Virtualización: WMWare 5.6</p> <p>S.O Virtual 1: Windows 2003 Server</p> <p>S.O Virtual 2: Linux Centos 5.5</p>

Elaborado por: Autora de Tesis

3.2.6 Paso 4: Realizar las Pruebas

Una vez explicado los pasos previos, se mide el rendimiento de manera objetiva usando los programas:

Cliente Windows

- Webserver stress Tool 7
- Colasoft Capsa

Cliente Linux

- Comandos httpperf, httping, ab

Los mismo que serán instalados en las maquinas clientes dentro de la red y que harán el trabajo de poner a prueba los servidores.

3.2.6.1 Procedimiento

Las pruebas de rendimiento son un componente crítico de un sitio web o el éxito global de la aplicación web. Desde un punto de vista de rendimiento, su objetivo es asegurarse de que el usuario final o cliente haga clic en el ratón y se cumpla a tiempo su petición.

Se debe optimizar el servidor web, para que el 95% de todas las peticiones web sean procesadas en menos de 10 segundos.

Tabla III.V. Umbrales recomendados de rendimiento para su sitio web o aplicación web.

Tiempo de descarga	Vista de Usuario
<0,1 s	Usuario siente que el sistema está reaccionando instantáneamente.
<1,0s	La experiencia del usuario no se vea comprometida. Aunque el usuario no está satisfecho con la espera, que aún se centran en la página web actual.
<10 s	Como los tiempos de espera acercarse a 10 segundos, los estudios han demostrado que la posibilidad de distracción de usuario aumenta considerablemente
> 10s	El usuario está más probable es que distrae de la actual página web y pierde interés.

Elaborado por: Autora de Tesis

Las pruebas de rendimiento tienen una serie de consideraciones técnicas para garantizar que esas cuestiones se resuelven dentro de los negocios. Por ejemplo, se considera las siguientes preguntas:

¿Está preparado el servidor web para el tráfico que el usuario está esperando?

¿Está preparado su servidor web para soportar el aumento de los visitantes en los meses y años por venir?

¿Puede sobrevivir su servidor web un aumento masivo en el tráfico de usuario (por ejemplo, si su sitio web se menciona en la televisión nacional o de su correo electrónico un boletín de la empresa a todos los clientes actuales y potenciales)?

¿Cuántos usuarios pueden gestionar su servidor web para que los usuarios empiezan a recibir mensajes de error o tiempos de espera del servidor?

¿Cuántos segundos le toma a un visitante de su sitio web para recibir una página tras hacer clic en un enlace? En condiciones normales? Bajo condiciones fuertes?

¿Están los scripts y bases de datos optimizados para funcionar tan pronto como sea posible y cómo interactúan entre sí bajo cargas pesadas hacia el servidor web?

¿Está el servicio de alojamiento web haciendo un buen trabajo?

3.2.6.2 Configuración de Herramientas para Medición

Las herramientas utilizadas para la medición son: Colasoft, Web Server Stress Tools.

3.2.6.2.1 Configuración de Colasoft , Cliente Windows

a. Configuración de la Tarjeta de Red para Análisis de Tráfico

Cada vez que se inicia una nueva captura, se requiere seleccionar el adaptador. Adaptadores disponibles se presentan en la configuración del proyecto - pestaña "Adapter", para este caso se elige uno. Una descripción acerca de este adaptador está mostrada en el panel de propiedades inferior.

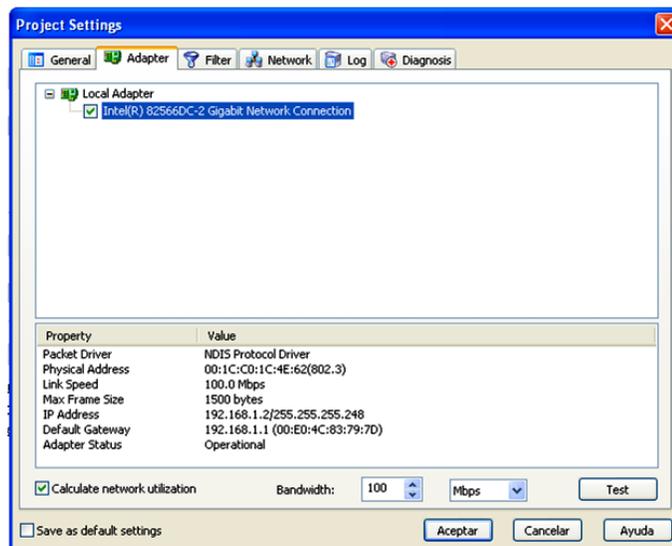


Figura III.13. Adaptador para pruebas de tráfico de Red

b. Configuración de opciones de Red

Se configura el perfil de red. En la pestaña “Network” de la configuración del proyecto, se personaliza los parámetros físicos o puntos finales de red, por lo tanto se asignan los nodos de la red que intervienen en la simulación.

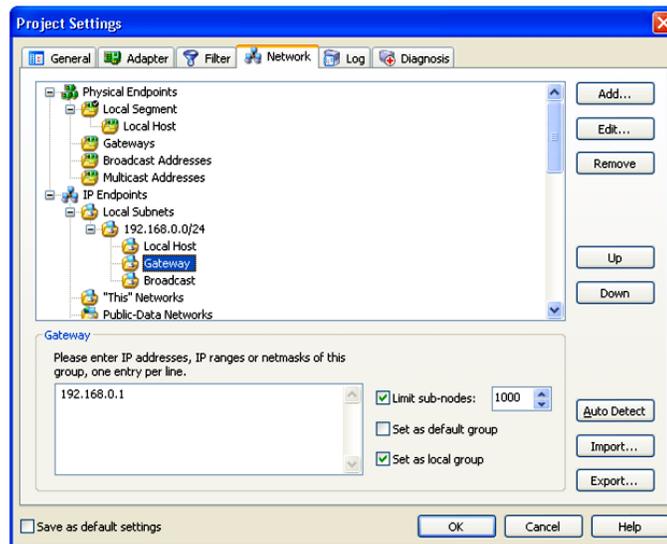


Figura III.14. Configuración de Endpoints

c. Configuración del Analizador de Diagnostico

Se diagnostica la Red desde los paquetes capturados y se lista todos los eventos con un nivel de seguridad, con la posibilidad de no solo ver los eventos en la Capa de Red sino todos los relacionados con información como origen, destino, número de paquetes. El nivel de seguridad indica si se dan errores menores o críticos.

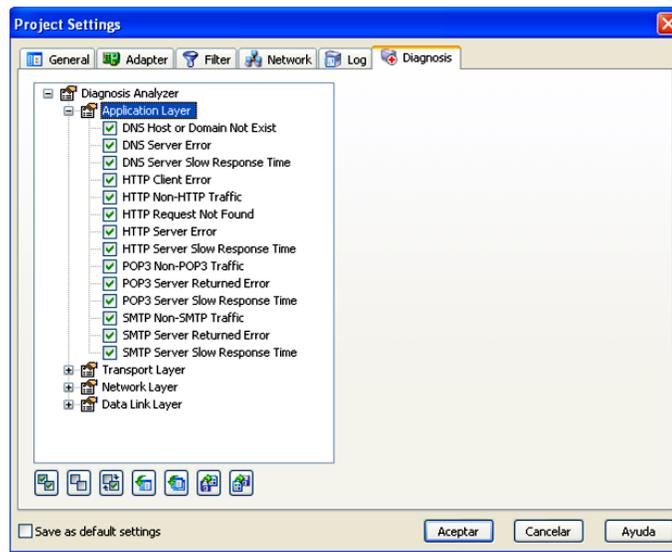


Figura III.15. Configuración de eventos para la capa de Aplicación

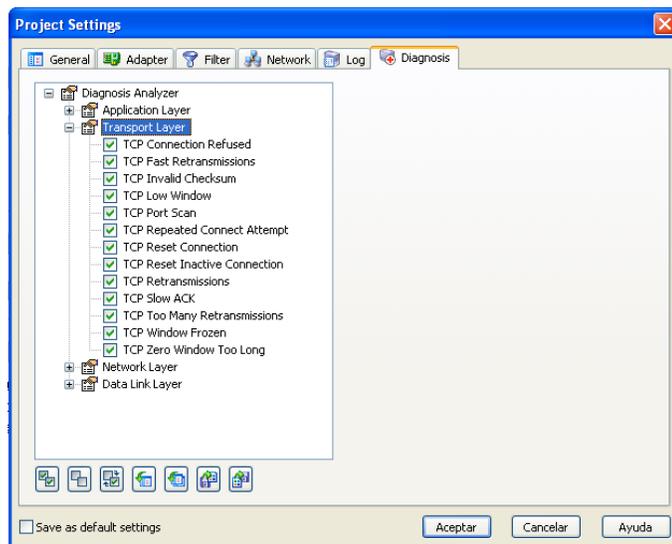


Figura III.16. Configuración de eventos para la capa de Transporte

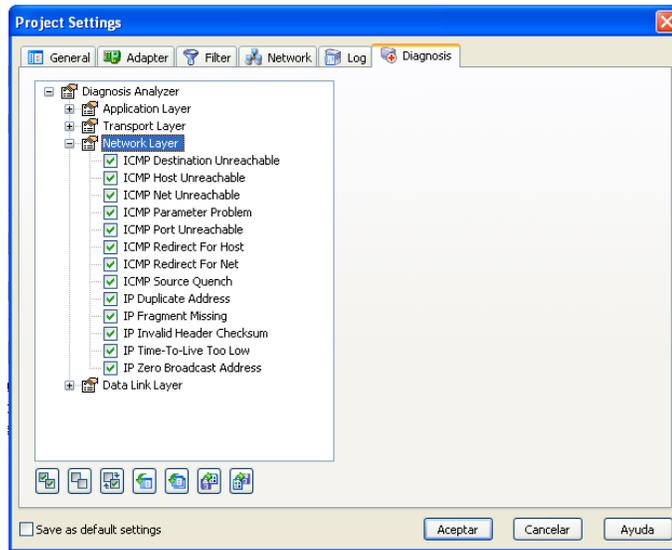


Figura III.17. Configuración de eventos para la capa de Red

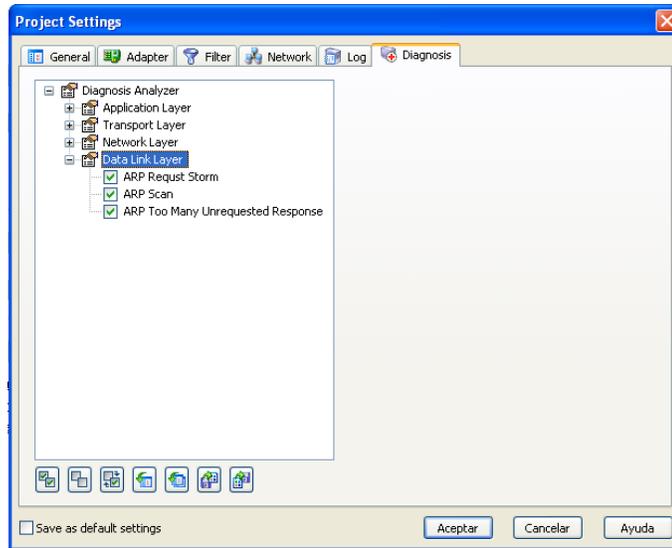


Figura III.18. Configuración de eventos para la capa de Enlace de Datos

3.2.6.2.2 Configuración de Web Stress Tools

Las pruebas de estrés son simulados como "fuerza bruta", son los ataques que se aplican una carga excesiva a un servidor web. En el mundo real este tipo de situación pueden ser creadas por un aumento masivo de los usuarios, causadas por un referente de gran tamaño.

Otro ejemplo sería una campaña de marketing por correo electrónico enviado a los clientes potencial es que les pide que vaya a la página web para registrarse en un servicio o solicitar información adicional. Una negación involuntaria deservicio a los clientes potenciales que están listos para aprender más acerca de su producto podría tener un grave impacto en su balance final.

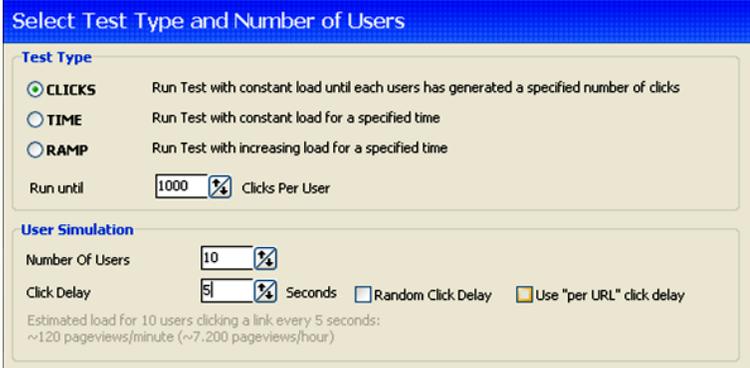
El propósito de estas pruebas de esfuerzo o estrés es el de estimarla carga máxima que puede soportarlos servidores web.

a. Peticiones por Clic

El número de usuarios puede ser un valor entre 1 y 10.000. Pero tomando en cuenta que el número máximo de usuarios simultáneos que pueden ser simulados con éxito depende de la potencia de computación de la máquina cliente que ejecuta el Webserver Stress Tool.

Se muestra además el estado de la carga del CPU cliente en la barra de estado. Si la máquina cliente se ejecuta con carga del 100%, indica que se ha alcanzado el límite de procesamiento de la máquina.

A continuación, se configura el intervalo de tiempo de retardo entre click para los usuarios simulados. Este ajuste es tan importante como el número de usuarios. Cuanto menor sea el tiempo de retardo entre clics, mayor es el nivel de estrés en el servidor web.



Select Test Type and Number of Users

Test Type

CLICKS Run Test with constant load until each users has generated a specified number of clicks

TIME Run Test with constant load for a specified time

RAMP Run Test with increasing load for a specified time

Run until Clicks Per User

User Simulation

Number Of Users

Click Delay Seconds Random Click Delay Use "per URL" click delay

Estimated load for 10 users clicking a link every 5 seconds:
~120 pageviews/minute (~7,200 pageviews/hour)

Figura III.19. Configuración de peticiones o clicks para estresar los servidores web IIS y Apache

Las direcciones web que serán analizadas generando las peticiones en cada una de ellas, con un intervalo de tiempo entre peticiones de 5 segundos.

Select URLs or Edit URL Script						
URL Pattern						
<input checked="" type="radio"/> Simple URL Sequence with <input type="text" value="4"/> URLs <input type="radio"/> Custom URL Script						
<input type="button" value="Add URL"/> <input type="button" value="Delete URL"/> <input type="button" value="Clear URLs"/> <input type="button" value="URL Recorder"/> <input type="button" value="Data Merging"/>						
URL #	Name	Click Delay [s]	URL	POST data (or @filename@)	Username	Password
1	1	5	http://www.serveriis.com/			
2	2	5	http://www.serveriis.com/Services.html			
3	3	5	http://www.serveriis.com/Products.html			
4	4	5	http://www.serveriis.com/Contacts.html			

Figura III.20. Direcciones Web del servidor IIS que serán analizadas

Select URLs or Edit URL Script						
URL Pattern						
<input checked="" type="radio"/> Simple URL Sequence with <input type="text" value="4"/> URLs <input type="radio"/> Custom URL Script						
<input type="button" value="Add URL"/> <input type="button" value="Delete URL"/> <input type="button" value="Clear URLs"/> <input type="button" value="URL Recorder"/> <input type="button" value="Data Merging"/>						
URL #	Name	Click Delay [s]	URL	POST data (or @filename@)	Username	Password
1	1	5	http://www.serverapache.com/			
2	2	5	http://www.serverapache.com/Services.html			
3	3	5	http://www.serverapache.com/Products.html			
4	4	5	http://www.serverapache.com/Contacts.html			

Figura III.21. Direcciones Web del servidor Apache que serán analizadas

b. Peticiones por tiempo con carga constante

El análisis de Tráfico de red Las pruebas que se ejecutan durante un número determinado de minutos. Una prueba cronometrada se utiliza a menudo para "quemar en las pruebas", por ejemplo, para mantener un servidor a plena carga durante 10 horas.

Estas pruebas de carga constante se llevan a cabo poniendo a prueba los sitios web utilizando la mejor estimación del tráfico. Se considera esto como una "prueba en el mundo real" de la página web.

El primer paso es definir el tiempo máximo que debe tomar (desde un punto de vista de la usabilidad y la experiencia del cliente, no un técnico) para que una página se cargue. Una buena regla de oro es para asegurarse de que ningún visitante del sitio web espera más de diez (10) segundos para una página web para cargar.

The screenshot shows a configuration window with a blue header. Under 'Test Type', three radio buttons are present: 'CLICKS' (unselected), 'TIME' (selected), and 'RAMP' (unselected). Below this, a 'Run Test For' field is set to '2' minutes, with a timestamp '(from 22/02/2012 23:51:55 until 22/02/2012 23:53:55)'. The 'User Simulation' section has 'Number Of Users' set to '10' and 'Click Delay' set to '5' seconds. There are two checkboxes: 'Random Click Delay' (unchecked) and 'Use "per URL" click delay' (unchecked). At the bottom, it states: 'Estimated load for 10 users clicking a link every 5 seconds: ~120 pageviews/minute (~7,200 pageviews/hour)'.

Figura III.22. Configuración de tiempo para pruebas de carga constante

c. Peticiones por tiempo con carga variable

Las pruebas denominada Ramp son variaciones de las pruebas de estrés en el que se incrementa el número de usuarios durante la vida útil de la prueba-a partir de un solo usuario a cientos de usuarios.

Select Test Type and Number of Users

Test Type

- CLICKS** Run Test with constant load until each users has generated a specified number of clicks
- TIME** Run Test with constant load for a specified time
- RAMP** Run Test with increasing load for a specified time

Run Test For Minutes (from 22/02/2012 23:53:13 until 22/02/2012 23:55:13)

User Simulation

Number Of Users

Click Delay Seconds Random Click Delay Use "per URL" click delay

Estimated load for 10 users clicking a link every 5 seconds:
~120 pageviews/minute (~7.200 pageviews/hour)

Figura III.23. Configuración de tiempo para pruebas de carga variable

CAPITULO IV

ANALISIS DE RESULTADOS

4.1 INTRODUCCION

En este capítulo se describe los resultados obtenidos y su análisis correspondiente.

La evaluación de las características de cada servidor web tiene un peso asignado es del 20% del valor total de la comparativa, y el análisis de los resultados de las pruebas realizadas cuyos valores pueden ser apreciados en forma porcentual tienen un peso del 80% del valor total.

Los métodos, variables utilizadas se han descrito en el Capítulo III – Comparativa de Servidores.

4.2 EVALUACIÓN DE LAS CARACTERÍSTICAS

Observamos las características que cada servidor dispone, donde el 20% del valor total de la comparativa es asignado al servidor(s), cabe indicar que todos los servidores soportan conexión a Base de datos.

El resultado de esta prueba se realizara con la asignación de pesos de la siguiente forma:

Pesos

Si = 1

No = 0

Tabla IV.VI.Características de cada servidor

	Apache	IIS
Características generales		
Código Abierto	Si	No
Compatibilidad con Windows	Si	Si
Linux	Si	No
Seguridad		
Autenticación	Si	Si
Https	Si	Si

Virtual Hosts	Si	Si
Lenguajes del lado del servidor		
Php	Si	Si
Perl	Si	Si
Python	Si	Si
Asp	Si	Si
Porcentaje	10	8

Elaborado por: Autora de Tesis

10 *caracteristicas* → 100%

$$10 \quad x \quad = \quad 100\%$$

$$8 \quad x \quad = \quad 80\%$$

100 → 20%

$$100 \quad x \quad = \quad 20\%$$

$$80 \quad x \quad = \quad 16\%$$

Valor Alcanzado	
Apache	20%
IIS	16%

4.3 GRAFICAS Y RESULTADOS DE LAS PRUEBAS

4.3.1 Prueba 1. Análisis Tráfico de red

Número de Usuarios: 10 simultáneos

Número de Peticiones por usuario: 1000

Traffic	Bytes	Packets	Utilization	bps	pps
Total Traffic	76.176 MB	143,290	0.001%	1.248 ...	2
Broadcast Traffic Sent	10.279 KB	84	0.000%	0 bps	0
Multicast Traffic Sent	8.938 KB	143	0.000%	0 bps	0

Figura IV.24. Tráfico broadcast, multicast y total generado entre el cliente y el servidor IIS

Traffic	Bytes	Packets	Utilization	bps	pps
Total Traffic	75.567 MB	151,281	0.001%	1.248 ...	2
Broadcast Traffic Sent	11.714 KB	97	0.000%	0 bps	0
Multicast Traffic Sent	8.938 KB	143	0.000%	0 bps	0

Figura IV.25. Tráfico broadcast, multicast y total generado entre el cliente y el servidor

Apache

TCP Packets					
	Bytes	Packets	Utilization	bps	pps
TCP SYN Packets	1.248 MB	19,822	0.000%	0 bps	0
TCP FIN Packets	7.894 MB	19,834	0.000%	0 bps	0
TCP Reset Packets	0 B	0	0.000%	0 bps	0
TCP Connections					Count
TCP Connection Initiated					9,917
TCP Connection Established					9,917
DNS Analysis					Count
DNS Request					277
DNS Correct Response					15
DNS Error Response					28

Figura IV.26. Total de Paquetes TCP y Peticiones DNS cliente-servidor IIS

TCP Packets					
	Bytes	Packets	Utilization	bps	pps
TCP SYN Packets	1.268 MB	20,148	0.000%	0 bps	0
TCP FIN Packets	8.005 MB	20,150	0.000%	0 bps	0
TCP Reset Packets	0 B	0	0.000%	0 bps	0
TCP Connections					Count
TCP Connection Initiated					10,075
TCP Connection Established					10,075
DNS Analysis					Count
DNS Request					308
DNS Correct Response					21
DNS Error Response					24

Figura IV.27. Total de Paquetes TCP y Peticiones DNS cliente-servidor Apache

HTTP Analysis		Count
HTTP Connections		9,910
HTTP Requests		9,961
HTTP Servers Seen		1

Figura IV.28. Peticiones y Conexiones HTTP que se establecen cliente-servidor IIS

HTTP Analysis		Count
HTTP Connections		10,073
HTTP Requests		10,073
HTTP Servers Seen		1

Figura IV.29. Peticiones y Conexiones HTTP que se establecen cliente-servidor Apache

Name	Count
All Diagnosis Events	57,416
Application Layer	98
DNS Server Error	28
DNS Server Slow Response Time	14
HTTP Request Not Found	56
Transport Layer	57,176
TCP Invalid Checksum	57,176
Network Layer	142
IP Time-To-Live Too Low	142

Figura IV.30. Resultado de errores en peticiones DNS y HTTP (IIS)

Name	Count
All Diagnosis Events	54,260
Application Layer	134
DNS Server Error	24
DNS Server Slow Response Time	21
HTTP Request Not Found	89
Transport Layer	53,984
TCP Invalid Checksum	53,984
Network Layer	142
IP Time-To-Live Too Low	142

Figura IV.31. Resultado de errores en peticiones DNS y HTTP (Apache)

Name	Bytes	Packets	Bits Per Second	Bytes%	Packets%	Connections
Ethernet II	76.176 MB	143,290	1.248 Kbps	100.000%	100.000%	9,917
IP	76.175 MB	143,270	1.248 Kbps	99.998%	99.986%	9,917
TCP	75.894 MB	139,559	0 bps	99.630%	97.396%	9,917
HTTP	75.891 MB	139,538	0 bps	99.626%	97.382%	9,916
NetBIOS	3.078 KB	21	0 bps	0.004%	0.015%	1
Session Service	3.078 KB	21	0 bps	0.004%	0.015%	1
ICMP	241.617 KB	3,172	1.248 Kbps	0.310%	2.214%	0
Echo Reply	120.809 KB	1,586	624 bps	0.155%	1.107%	0
Echo Req	120.809 KB	1,586	624 bps	0.155%	1.107%	0
UDP	36.684 KB	397	0 bps	0.047%	0.277%	0
DNS	25.697 KB	320	0 bps	0.033%	0.223%	0
Query	22.004 KB	277	0 bps	0.028%	0.193%	0
Error	2.259 KB	28	0 bps	0.003%	0.020%	0
Response	1.435 KB	15	0 bps	0.002%	0.010%	0
NetBIOS	8.221 KB	68	0 bps	0.011%	0.047%	0
Name Service	5.262 KB	56	0 bps	0.007%	0.039%	0
Datagram Service	2.959 KB	12	0 bps	0.004%	0.008%	0
BOOTP	2.703 KB	8	0 bps	0.003%	0.006%	0
DHCP	2.703 KB	8	0 bps	0.003%	0.006%	0
Other	64 B	1	0 bps	0.000%	0.001%	0
IGMP	8.875 KB	142	0 bps	0.011%	0.099%	0
ARP	1.250 KB	20	0 bps	0.002%	0.014%	0
Request	832 B	13	0 bps	0.001%	0.009%	0
Response	448 B	7	0 bps	0.001%	0.005%	0

Figura IV.32. Análisis de Trafico por Protocolos (IIS)

Name	Bytes	Packets	Bits Per Second	Bytes%	Packets%	Connections
Ethernet II	75.567 MB	151,281	1.248 Kbps	100.000%	100.000%	10,075
IP	75.565 MB	151,252	1.248 Kbps	99.998%	99.981%	10,075
TCP	75.213 MB	146,576	0 bps	99.531%	96.890%	10,075
HTTP	75.210 MB	146,555	0 bps	99.527%	96.876%	10,074
NetBIOS	3.078 KB	21	0 bps	0.004%	0.014%	1
Session Service	3.078 KB	21	0 bps	0.004%	0.014%	1
ICMP	312.152 KB	4,098	1.248 Kbps	0.403%	2.709%	0
Echo Reply	156.076 KB	2,049	624 bps	0.202%	1.354%	0
Echo Req	156.076 KB	2,049	624 bps	0.202%	1.354%	0
UDP	39.808 KB	436	0 bps	0.051%	0.288%	0
DNS	28.926 KB	353	0 bps	0.037%	0.233%	0
Query	24.883 KB	308	0 bps	0.032%	0.204%	0
Error	2.033 KB	24	0 bps	0.003%	0.016%	0
Response	2.010 KB	21	0 bps	0.003%	0.014%	0
NetBIOS	8.319 KB	74	0 bps	0.011%	0.049%	0
Name Service	6.105 KB	65	0 bps	0.008%	0.043%	0
Datagram Service	2.214 KB	9	0 bps	0.003%	0.006%	0
BOOTP	2.500 KB	8	0 bps	0.003%	0.005%	0
DHCP	2.500 KB	8	0 bps	0.003%	0.005%	0
Other	64 B	1	0 bps	0.000%	0.001%	0
IGMP	8.875 KB	142	0 bps	0.011%	0.094%	0
ARP	1.813 KB	29	0 bps	0.002%	0.019%	0
Request	1.000 KB	16	0 bps	0.001%	0.011%	0
Response	832 B	13	0 bps	0.001%	0.009%	0

Figura IV.33. Análisis de Trafico por Protocolos (Apache)

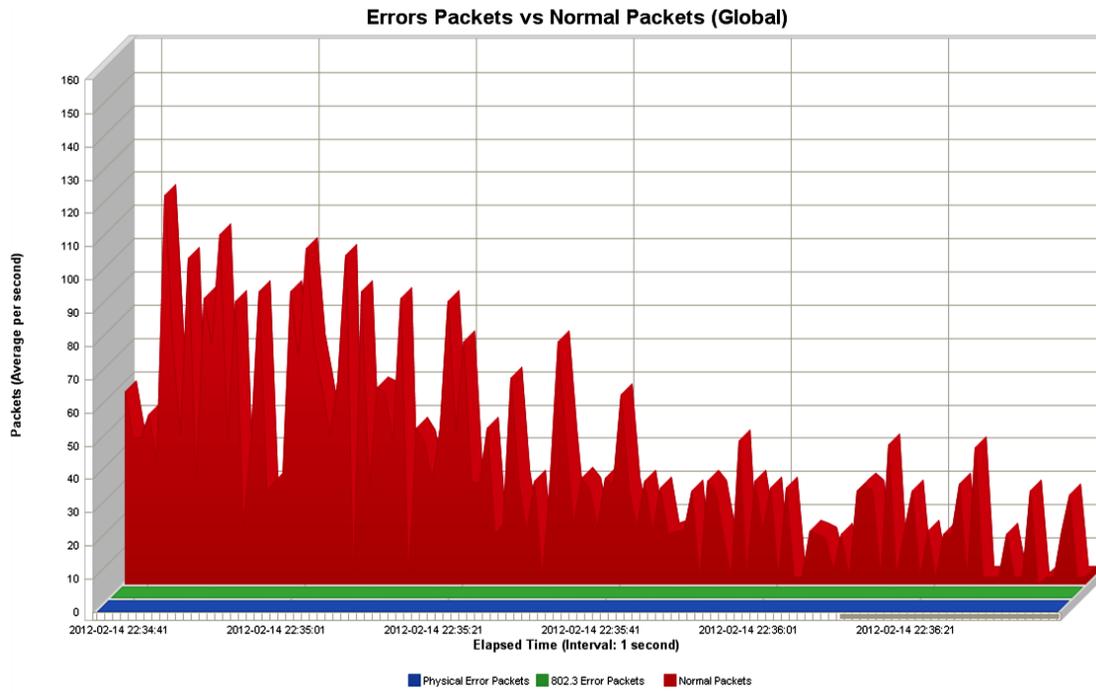


Figura IV.34. Relación de paquetes con error y paquetes normales intercambiados (ISS)

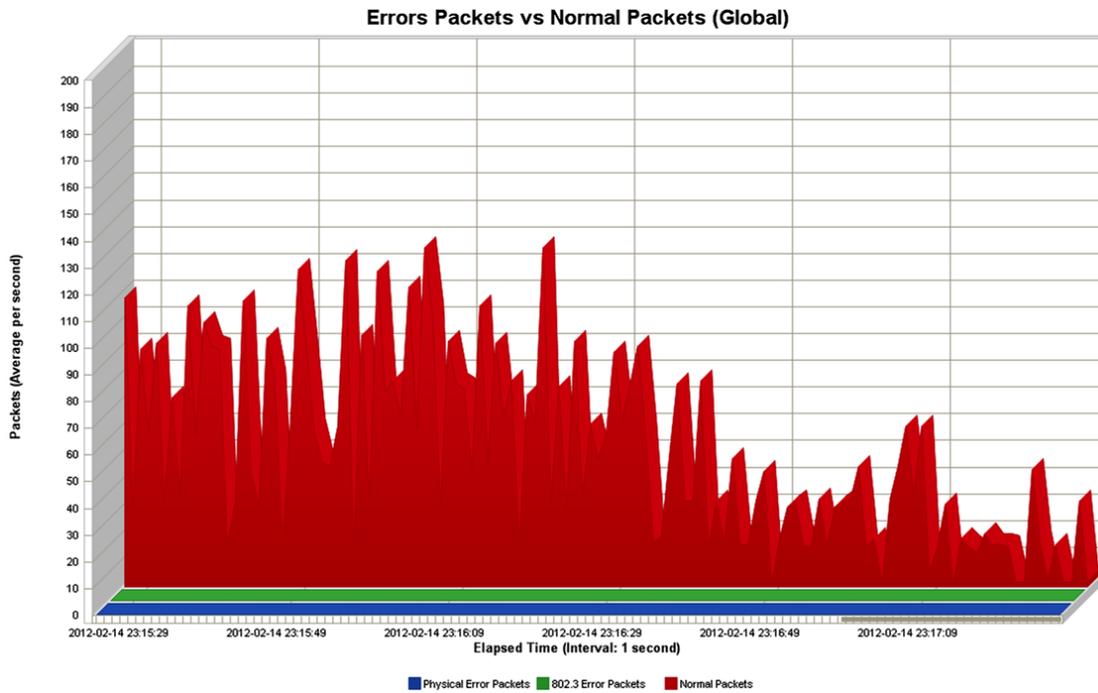


Figura IV.35. Relación de paquetes con error y paquetes normales intercambiados (Apache)

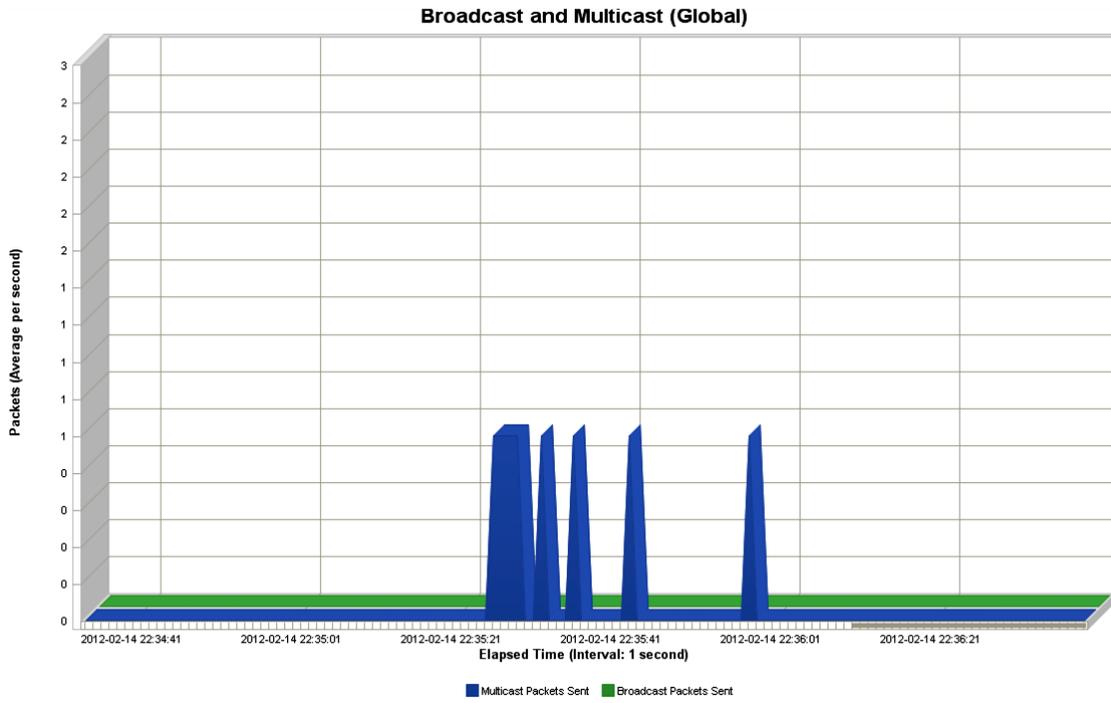


Figura IV.36. Relación de paquetes broadcast y multicast enviados (ISS)

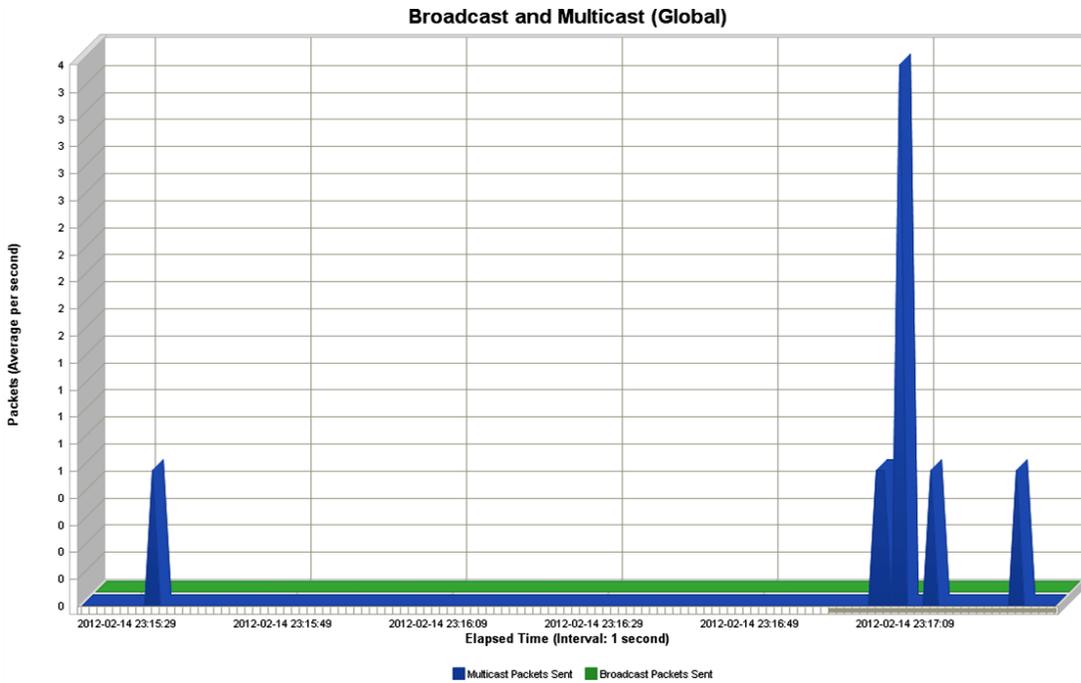


Figura IV.37. Relación de paquetes broadcast y multicast enviados (Apache)

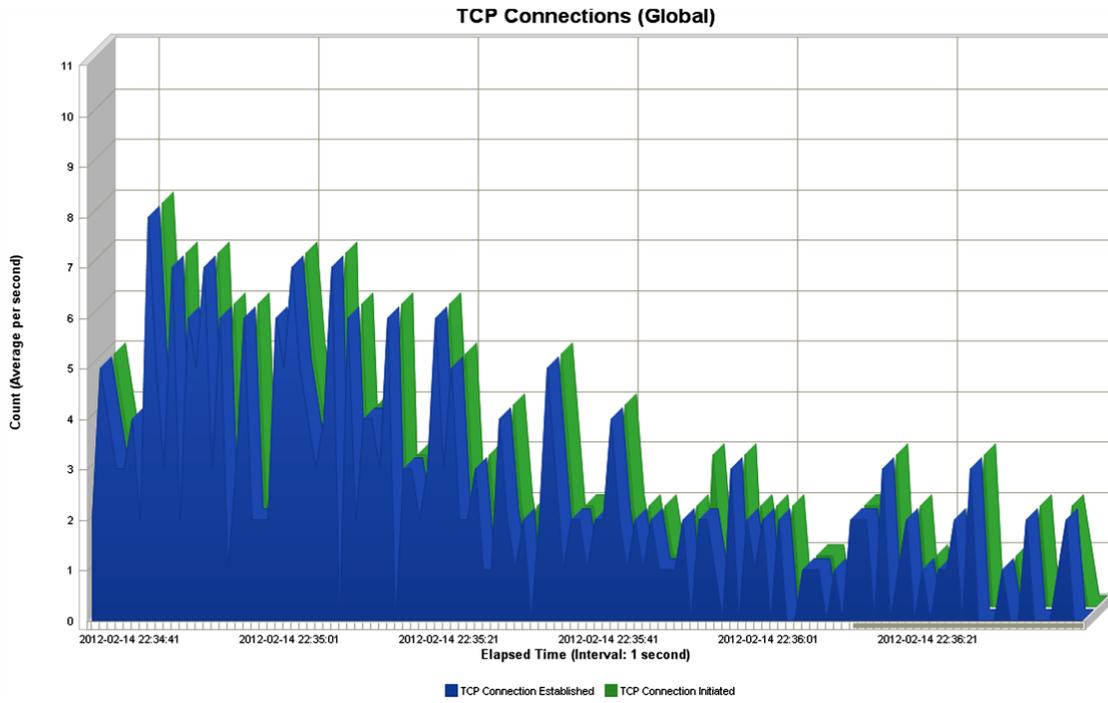


Figura IV.38. Relación de Conexiones TCP iniciadas y establecidos (ISS)

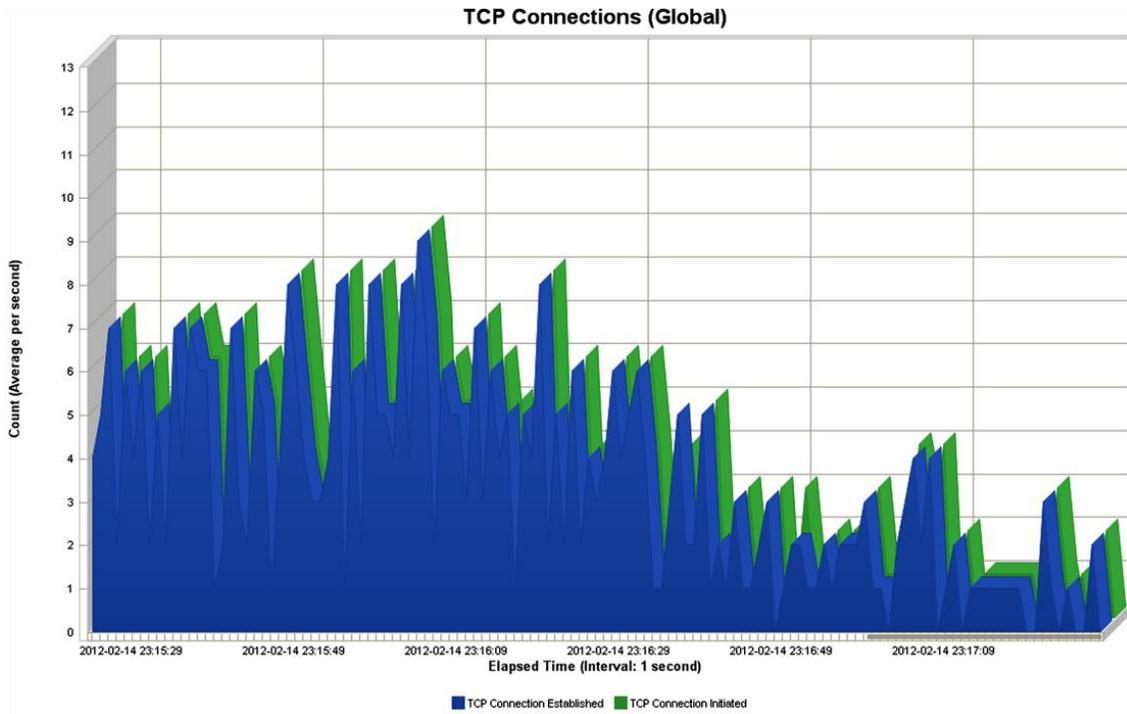


Figura IV.39. Relación de Conexiones TCP iniciadas y establecidos (Apache)

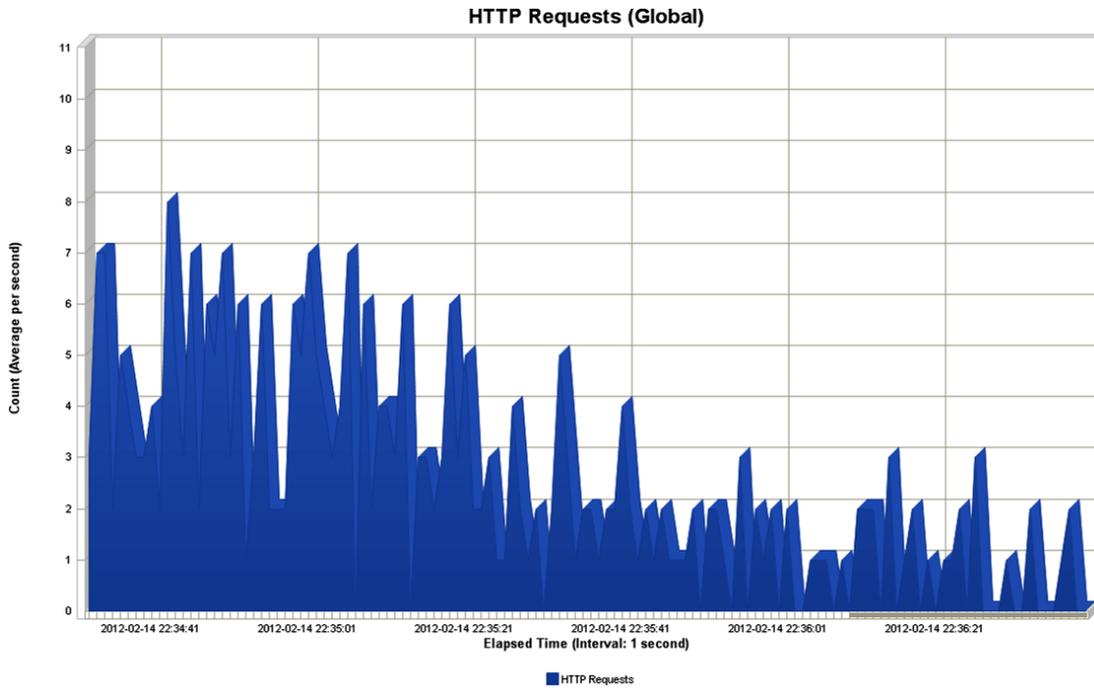


Figura IV.40. Paquetes HTTP promedio por segundo (ISS)

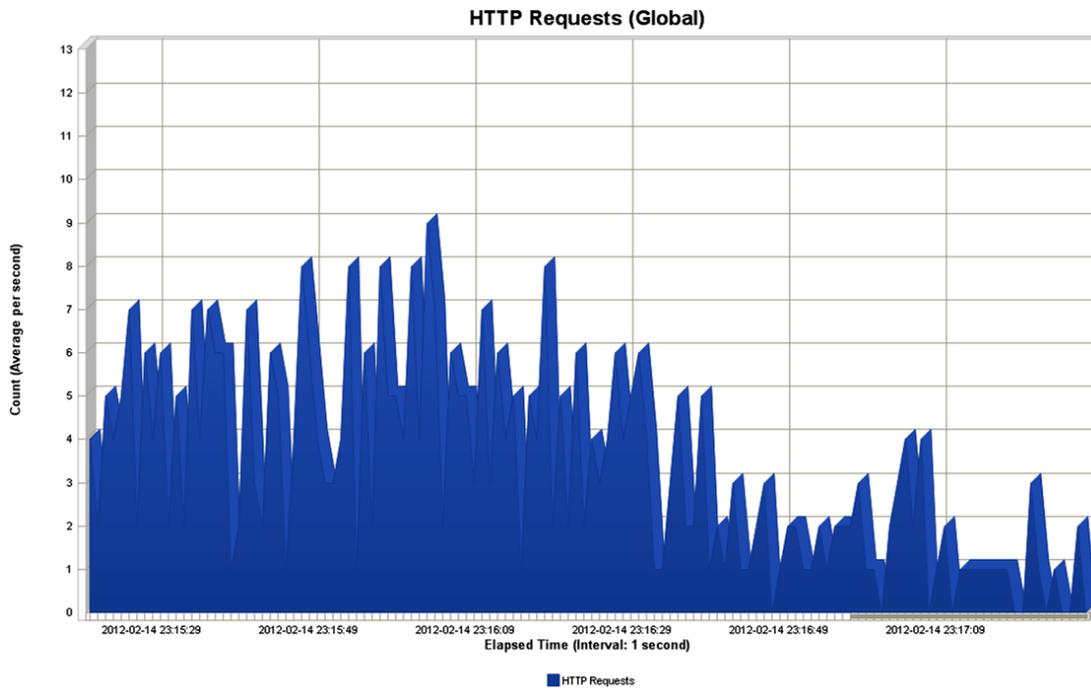


Figura IV.41. Paquetes HTTP promedio por segundo (Apache)

4.3.1.1 PRUEBA 1.1 PING CON CARGA, PING ICMP

```

C:\WINDOWS\system32\cmd.exe
Respuesta desde 192.168.1.5: bytes=32 tiempo<1m TTL=128
Respuesta desde 192.168.1.5: bytes=32 tiempo=2ms TTL=128
Respuesta desde 192.168.1.5: bytes=32 tiempo<1m TTL=128
Estadísticas de ping para 192.168.1.5:
  Paquetes: enviados = 1785, recibidos = 1785, perdidos = 0
  (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
  Mínimo = 0ms, Máximo = 166ms, Media = 0ms
Control-C
^C
C:\Documents and Settings\Administrador>

```

Figura IV.42. Respuesta de tiempos de ida y vuelta mientras se ejecuta 1000 peticiones por 10 usuarios hacia el servidor ISS

```

C:\WINDOWS\system32\cmd.exe
Respuesta desde 192.168.1.6: bytes=32 tiempo<1m TTL=64
Estadísticas de ping para 192.168.1.6:
  Paquetes: enviados = 2048, recibidos = 2048, perdidos = 0
  (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
  Mínimo = 0ms, Máximo = 79ms, Media = 0ms
Control-C
^C
C:\Documents and Settings\Administrador>

```

Figura IV.43. Respuesta de tiempos de ida y vuelta mientras se ejecuta 1000 peticiones por 10 usuarios hacia el servidor Apache

4.3.2 PRUEBA 2. PRUEBA CON CARGA CONSTANTE POR 2 MINUTOS CON 10 USUARIOS SIMULTÁNEOS

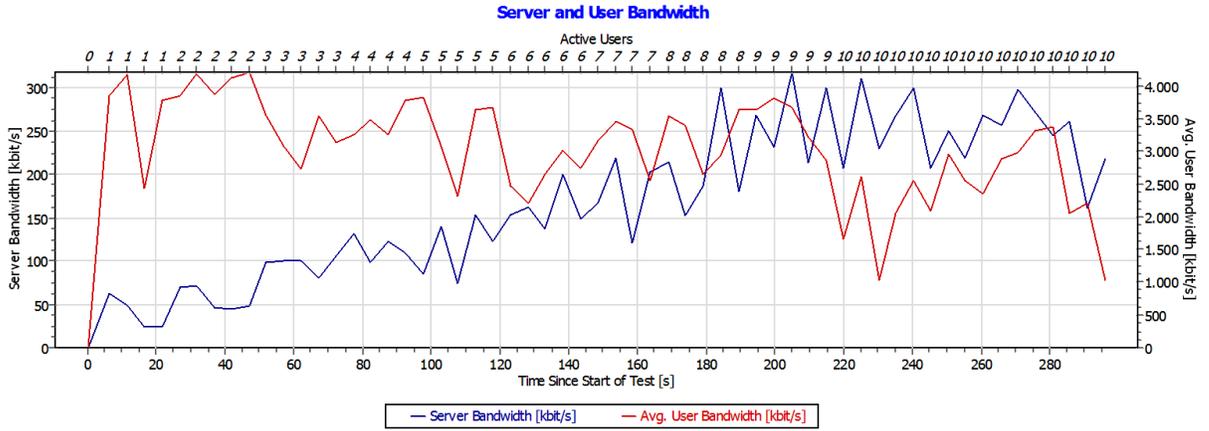


Figura IV.44. Promedio de Utilización de Ancho de banda cliente-servidor IIS

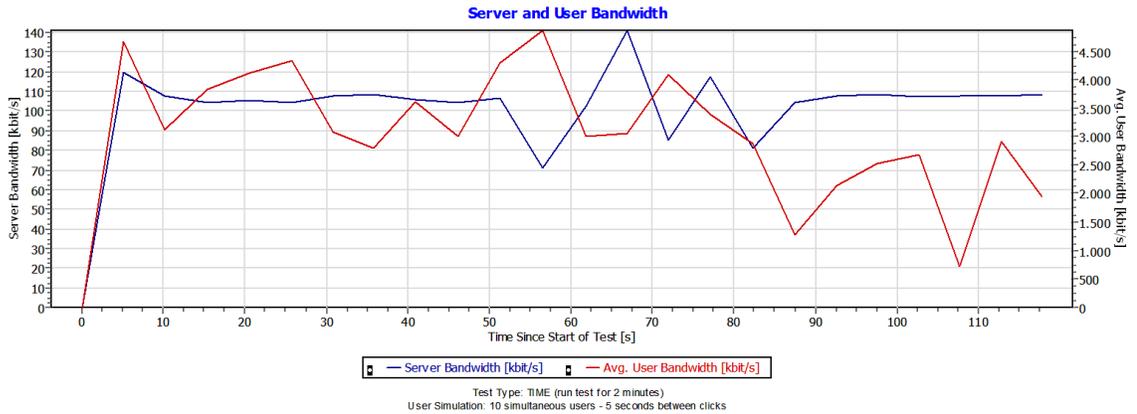


Figura IV.45. Promedio de Utilización de Ancho de banda cliente-servidor Apache

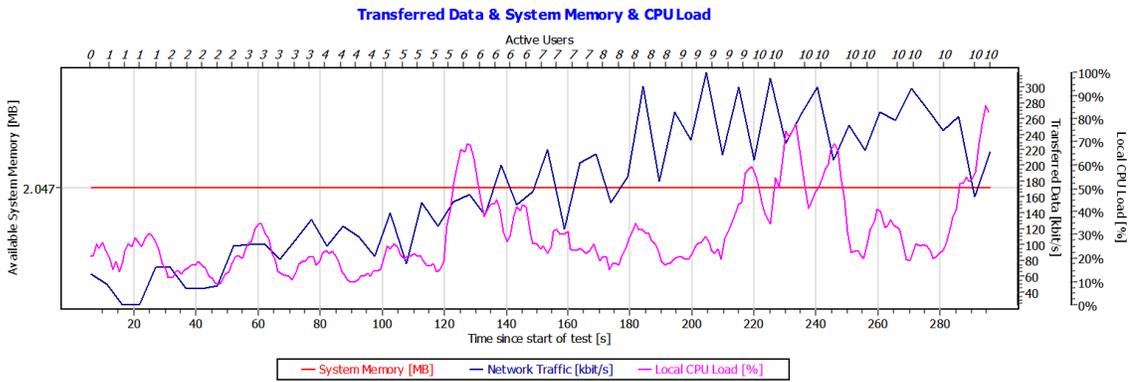


Figura IV.46. Memoria del sistema disponible, Trafico de Red y uso del CPU local (ISS)

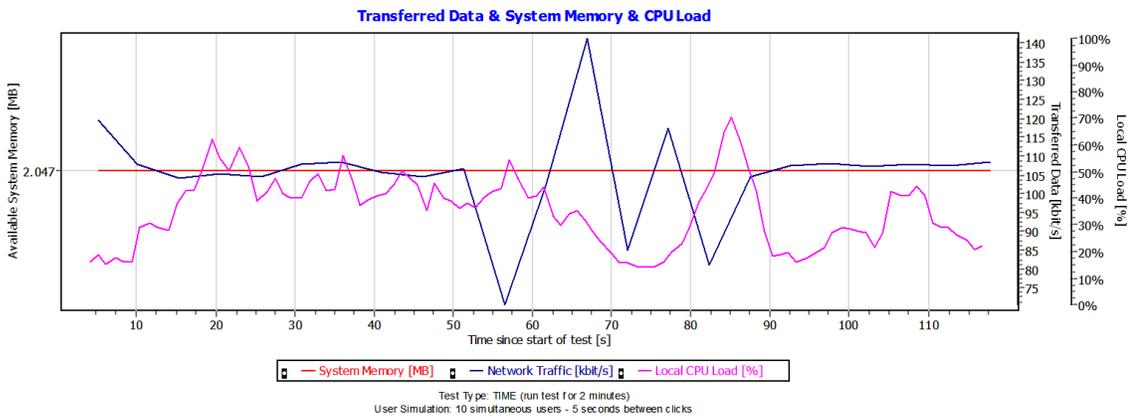


Figura IV.47. Memoria del sistema disponible, Trafico de Red y uso del CPU local (Apache)

4.3.3 PRUEBA 3. PRUEBA CARGA VARIABLE POR 2 MINUTOS CON 10 USUARIOS SIMULTÁNEOS

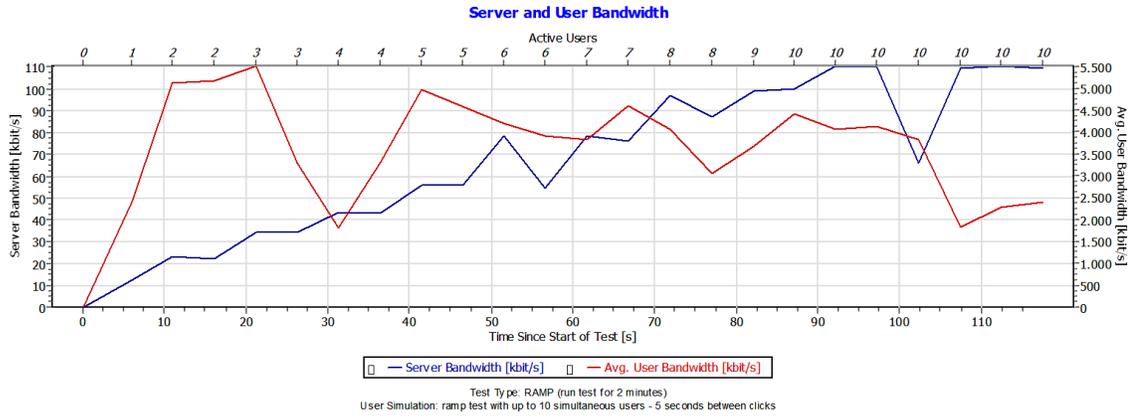


Figura IV.48. Ancho de banda promedio utilizado cliente-servidor IIS

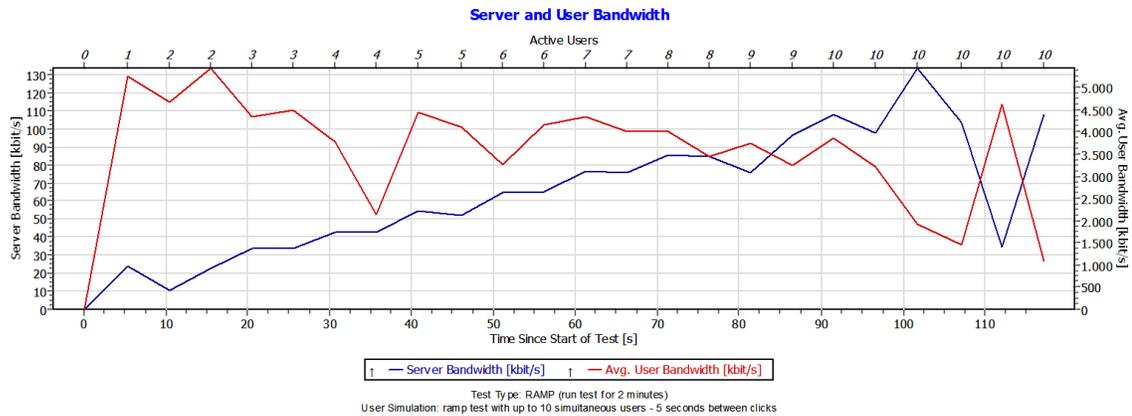


Figura IV.49. Ancho de banda promedio utilizado cliente-servidor Apache

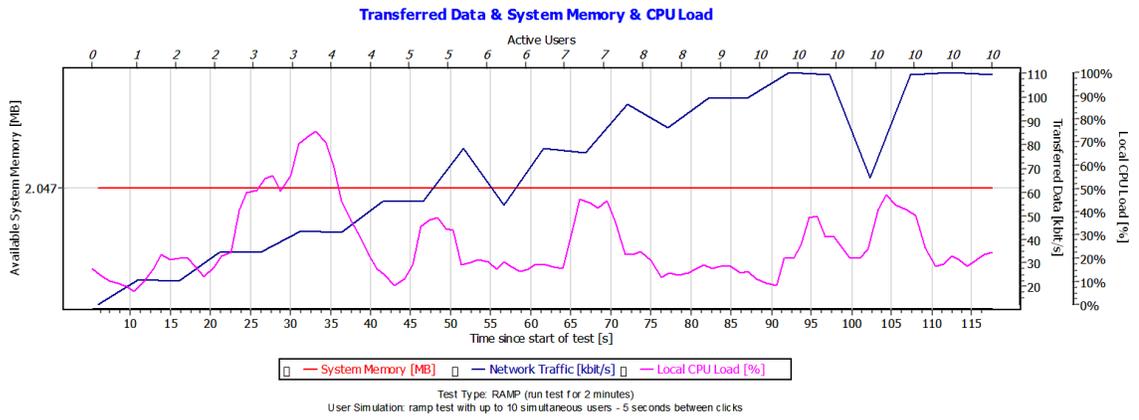


Figura IV.50. Memoria del sistema disponible, Trafico de Red y uso del CPU local (IIS)

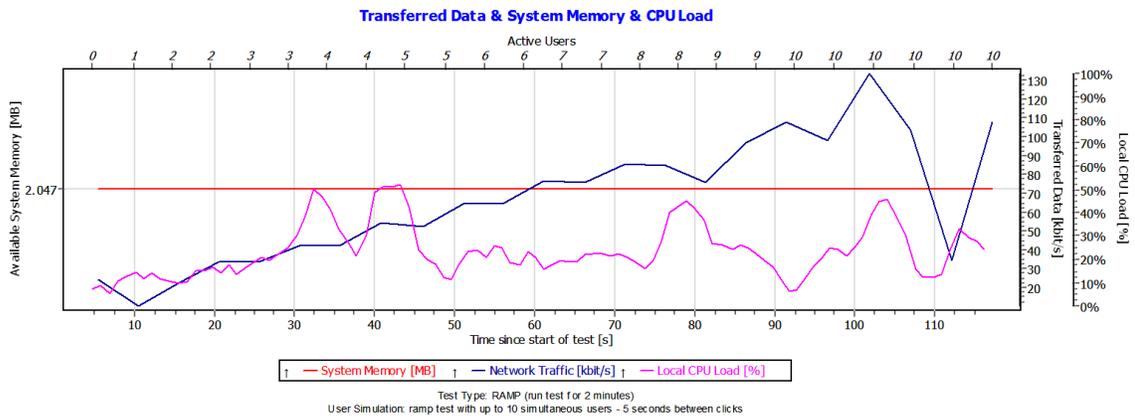


Figura IV.51. Memoria del sistema disponible, Trafico de Red y uso del CPU local

(Apache)

4.3.4 PRUEBA 4. LATENCIA, PING IP

Es el tiempo o lapso necesario para que un paquete de información se transfiera de un lugar a otro. Para ello, se usa el comando `htping` en el cliente Linux y se configura las peticiones como sigue:

```
#htping -c 5 -Gbg http:www.serveriis.com
```

```
#htping -c 5 -Gbg http:www.apache.com
```

El parámetro (-c 5) indican el número de conexiones simultaneas, y (-Gbg) indica que realice un Get al recibir el Head de la pagina web, la url q se analiza y la velocidad de transferencia.

Sin carga, significa el servidor esta en operación normal sin intervención del software Webserver Stress Tools.

Con carga, significa el servidor a parte de las peticiones que se le hace con el comando, está bajo estrés de webser stress tools con una prueba de tiempo con carga variable por dos minutos.

4.3.4.1 LATENCIA (SIN CARGA)

```
[root@server ~]# httping -c 5 -Gbg http://www.serveriis.com
PING www.serveriis.com:80 (http://www.serveriis.com):
connected to 192.168.1.5:80 (300 bytes), seq=0 time=33.38 ms 262041K
B/s
connected to 192.168.1.5:80 (300 bytes), seq=1 time=6.31 ms 3926KB/s
connected to 192.168.1.5:80 (300 bytes), seq=2 time=7.07 ms 2446KB/s
connected to 192.168.1.5:80 (300 bytes), seq=3 time=8.05 ms 2379KB/s
connected to 192.168.1.5:80 (300 bytes), seq=4 time=6.71 ms 2761KB/s
--- http://www.serveriis.com ping statistics ---
5 connects, 5 ok, 0.00% failed
round-trip min/avg/max = 6.3/12.3/33.4 ms
Transfer speed: min/avg/max = 2379/54711/262041 KB
[root@server ~]# █
```

a.

```
[root@server ~]# httping -c 5 -Gbg http://www.serverapache.com
PING www.serverapache.com:80 (http://www.serverapache.com):
connected to 192.168.1.6:80 (265 bytes), seq=0 time=22.82 ms 288870K
B/s
connected to 192.168.1.6:80 (265 bytes), seq=1 time=10.74 ms 1330KB/
s
connected to 192.168.1.6:80 (265 bytes), seq=2 time=11.65 ms 1225KB/
s
connected to 192.168.1.6:80 (265 bytes), seq=3 time=9.93 ms 1514KB/s
connected to 192.168.1.6:80 (265 bytes), seq=4 time=13.02 ms 938KB/s
--- http://www.serverapache.com ping statistics ---
5 connects, 5 ok, 0.00% failed
round-trip min/avg/max = 9.9/13.6/22.8 ms
Transfer speed: min/avg/max = 938/58775/288870 KB
[root@server ~]# █
```

b.

Figura IV.52. Latencia y tiempos de ida y vuelta, prueba sin carga a.) Servidor IIS, b.)

Servidor Apache

4.3.4.2 LATENCIA (CON CARGA)

```
[root@server ~]# httping -c 5 -Gbg http://www.serveriis.com
PING www.serveriis.com:80 (http://www.serveriis.com):
connected to 192.168.1.5:80 (300 bytes), seq=0 time=57.88 ms 87464KB
/s
connected to 192.168.1.5:80 (300 bytes), seq=1 time=7.67 ms 2692KB/s
connected to 192.168.1.5:80 (300 bytes), seq=2 time=7.40 ms 3102KB/s
connected to 192.168.1.5:80 (300 bytes), seq=3 time=7.49 ms 1898KB/s
connected to 192.168.1.5:80 (300 bytes), seq=4 time=6.10 ms 2370KB/s
--- http://www.serveriis.com ping statistics ---
5 connects, 5 ok, 0.00% failed
round-trip min/avg/max = 6.1/17.3/57.9 ms
Transfer speed: min/avg/max = 1898/19505/87464 KB
[root@server ~]# █
```

a.

```
[root@server ~]# httping -c 5 -Gbg http://www.serverapache.com
PING www.serverapache.com:80 (http://www.serverapache.com):
connected to 192.168.1.6:80 (265 bytes), seq=0 time=46.74 ms 2798KB/
s
connected to 192.168.1.6:80 (265 bytes), seq=1 time=8.26 ms 2050KB/s
connected to 192.168.1.6:80 (265 bytes), seq=2 time=7.27 ms 2982KB/s
connected to 192.168.1.6:80 (265 bytes), seq=3 time=9.25 ms 1521KB/s
connected to 192.168.1.6:80 (265 bytes), seq=4 time=9.29 ms 1296KB/s
--- http://www.serverapache.com ping statistics ---
5 connects, 5 ok, 0.00% failed
round-trip min/avg/max = 7.3/16.2/46.7 ms
Transfer speed: min/avg/max = 1296/2129/2982 KB
[root@server ~]# █
```

b.

Figura IV.53. Latencia y tiempos de ida y vuelta, prueba con carga a.) Servidor IIS, b.)

Servidor Apache

4.3.5 PRUEBA 5. PRUEBA DE SOBRECARGA DE PETICIONES

Total de peticiones: 5000

Usuarios simultáneos: 5

```

Server Hostname:      www.serveriis.com
Server Port:         80

Document Path:       /
Document Length:     7562 bytes

Concurrency Level:   5
Time taken for tests: 21.310838 seconds
Complete requests:   5000
Failed requests:     0
Write errors:        0
Total transferred:   39310000 bytes
HTML transferred:    37810000 bytes
Requests per second: 234.62 [#/sec] (mean)
Time per request:    21.311 [ms] (mean)
Time per request:    4.262 [ms] (mean, across all concurrent requests)
Transfer rate:       1801.34 [Kbytes/sec] received

Connection Times (ms)
      min  mean[+/-sd] median  max

Connect:    0    1   1.4    1    21
Processing:  1   19   6.8   20   47
Waiting:    0   15   6.4   15   44
Total:      2   20   6.9   21   49

Percentage of the requests served within a certain time (ms)
 50%    21
 66%    25
 75%    26
 80%    27
 90%    29
 95%    31
 98%    33
 99%    35
100%    49 (longest request)

```

Fig a.

```

[root@server ~]# ab -n 5000 -c 5 http://www.serverapache.com/
This is ApacheBench, Version 2.0.40-dev <$Revision: 1.146 $> apache-2.0
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Copyright 2006 The Apache Software Foundation, http://www.apache.org/

Benchmarking www.serverapache.com (be patient)
Completed 500 requests
Completed 1000 requests
Completed 1500 requests
Completed 2000 requests
Completed 2500 requests
Completed 3000 requests
Completed 3500 requests
Completed 4000 requests
Completed 4500 requests

```

Finished 5000 requests

```

Server Software:      Apache/2.2.3
Server Hostname:     www.serverapache.com
Server Port:         80

Document Path:       /
Document Length:     7368 bytes

Concurrency Level:   5
Time taken for tests: 29.95384 seconds
Complete requests:   5000
Failed requests:     0
Write errors:        0
Total transferred:   38165000 bytes
HTML transferred:    36840000 bytes
Requests per second: 171.85 [#/sec] (mean)
Time per request:    29.095 [ms] (mean)
Time per request:    5.819 [ms] (mean, across all concurrent requests)
Transfer rate:       1280.96 [Kbytes/sec] received

Connection Times (ms)
      min  mean[+/-sd] median  max
Connect:    0    0  1.3    1    18
Processing:  2   27  73.2   17   3856
Waiting:    0   15  57.8   13   3850
Total:      3   28  73.3   18   3860

Percentage of the requests served within a certain time (ms)
 50%    18
 66%    23
 75%    27
 80%    29
 90%    35
 95%    53
 98%   192
 99%   283
100%  3860 (longest request)
You have new mail in /var/spool/mail/root
[root@server ~]# █

```

Figura b

Figura IV.54. Número de peticiones por segundo y tiempo por cada petición que soporta sin carga a. Servidor IIs y b. Servidor Apache

```

[root@server ~]# ab -n 5000 -c 5 http://www.serveriis.com/
This is ApacheBench, Version 2.0.40-dev <$Revision: 1.146 $> apache-2.0
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Copyright 2006 The Apache Software Foundation, http://www.apache.org/

Benchmarking www.serveriis.com (be patient)
Completed 500 requests
Completed 1000 requests
Completed 1500 requests
Completed 2000 requests
Completed 2500 requests
Completed 3000 requests
Completed 3500 requests
Completed 4000 requests
Completed 4500 requests
Finished 5000 requests

Server Software:      Microsoft-IIS/6.0
Server Hostname:     www.serveriis.com
Server Port:         80

Document Path:       /
Document Length:     7562 bytes

Concurrency Level:   5
Time taken for tests: 25.57618 seconds
Complete requests:   5000
Failed requests:     0
Write errors:        0
Total transferred:   39310000 bytes
HTML transferred:    37810000 bytes
Requests per second: 199.54 [#/sec] (mean)
Time per request:    25.058 [ms] (mean)
Time per request:    5.012 [ms] (mean, across all concurrent requests)
Transfer rate:       1531.99 [Kbytes/sec] received

Connection Times (ms)
      min  mean[+/-sd] median  max
Connect:    0    1   1.6     1    23
Processing:  4   22   4.9    23    45
Waiting:    1   18   5.3    19    40
Total:      5   24   4.6    24    46

Percentage of the requests served within a certain time (ms)
 50%    24
 66%    26
 75%    27
 80%    28
 90%    29
 95%    31
 98%    35
 99%    37
100%    46 (longest request)

```

Figura a

```

[root@server ~]# ab -n 5000 -c 5 http://www.serverapache.com/
This is ApacheBench, Version 2.0.40-dev <$Revision: 1.146 $> apache-2.0
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Copyright 2006 The Apache Software Foundation, http://www.apache.org/

Benchmarking www.serverapache.com (be patient)
Completed 500 requests
Completed 1000 requests
Completed 1500 requests
Completed 2000 requests
Completed 2500 requests
Completed 3000 requests
Completed 3500 requests
Completed 4000 requests
Completed 4500 requests
Finished 5000 requests

Server Software:      Apache/2.2.3
Server Hostname:     www.serverapache.com
Server Port:         80

Document Path:      /
Document Length:    7368 bytes

Concurrency Level:   5
Time taken for tests: 32.10954 seconds
Complete requests:   5000
Failed requests:     0
Write errors:        0
Total transferred:   38165000 bytes
HTML transferred:    36840000 bytes
Requests per second: 156.20 [#/sec] (mean)
Time per request:    32.011 [ms] (mean)
Time per request:    6.402 [ms] (mean, across all concurrent requests)
Transfer rate:       1164.29 [Kbytes/sec] received

Connection Times (ms)
      min  mean[+/-sd] median  max
Connect:    0    2  60.0    1   3002
Processing:  2   28  50.5   19    712
Waiting:    1   14  12.3   13    268
Total:      3   31  78.4   20   3040

Percentage of the requests served within a certain time (ms)
 50%    20
 66%    25
 75%    28
 80%    31
 90%    36
 95%    69
 98%   221
 99%   309
100%  3040 (longest request)
[root@server ~]# █

```

Figura b

Figura IV.55. Número de peticiones por segundo y tiempo por cada petición que soporta con carga a. Servidor IIS y b. Servidor Apache

4.3.6 PRUEBA 6. PRUEBA DE SOBRECARGA DE USUARIOS

Total de peticiones: 5000

Usuarios simultáneos: 500

```
[root@server ~]# ab -n 5000 -c 500 http://www.serveriis.com/
This is ApacheBench, Version 2.0.40-dev <$Revision: 1.146 $> apache-2.0
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Copyright 2006 The Apache Software Foundation, http://www.apache.org/

Benchmarking www.serveriis.com (be patient)
Completed 500 requests
Completed 1000 requests
Completed 1500 requests
apr_socket_recv: Connection reset by peer (104)
Total of 1780 requests completed
[root@server ~]# █
```

Figura IV.56. Resultado sin éxito de sobre carga de usuarios para el servidor IIS

```

[root@server ~]# ab -n 5000 -c 500 http://www.serverapache.com/
This is ApacheBench, Version 2.0.40-dev <$Revision: 1.146 $> apache-2.
0
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.ne
t/
Copyright 2006 The Apache Software Foundation, http://www.apache.org/

Benchmarking www.serverapache.com (be patient)
Completed 500 requests
Completed 1000 requests
Completed 1500 requests
Completed 2000 requests
Completed 2500 requests
Completed 3000 requests
Completed 3500 requests
Completed 4000 requests
Completed 4500 requests
Finished 5000 requests

Server Software:      Apache/2.2.3
Server Hostname:     www.serverapache.com
Server Port:         80

Document Path:       /
Document Length:     7368 bytes

Concurrency Level:   500
Time taken for tests: 24.436397 seconds
Complete requests:   5000
Failed requests:     0
Write errors:        0
Total transferred:   38165000 bytes
HTML transferred:    36840000 bytes
Requests per second: 204.61 [#/sec] (mean)
Time per request:    2443.640 [ms] (mean)
Time per request:    4.887 [ms] (mean, across all concurrent reques
Transfer rate:       1525.18 [Kbytes/sec] received

Connection Times (ms)
      min  mean[+/-sd] median  max
Connect:    0  749 2362.9     1 21005
Processing:  6 1044 1503.6    602 18647
Waiting:    1 1016 1491.4    594 18645
Total:     140 1794 3008.9    618 24036

Percentage of the requests served within a certain time (ms)
 50%    618
 66%    764
 75%   1214
 80%   2747
 90%   3741
 95%   7186
 98%  11484
 99%  17036
100% 24036 (longest request)

```

Figura IV.57. Resultado con éxito de sobre carga de usuarios para el servidor IIS

4.3.7 PRUEBA 7. PORCENTAJE DE USO DE CPU

```
[root@server ~]# httpperf --server=www.serveriis.com --rate=10 --num-conn
s=100
httpperf --client=0/1 --server=www.serveriis.com --port=80 --uri=/ --ra
te=10 --send-buffer=4096 --recv-buffer=16384 --num-conns=100 --num-cal
ls=1
Maximum connect burst length: 1

Total: connections 100 requests 100 replies 100 test-duration 9.909 s

Connection rate: 10.1 conn/s (99.1 ms/conn, <=1 concurrent connections
)
Connection time [ms]: min 5.3 avg 8.0 max 22.3 median 7.5 stddev 2.0
Connection time [ms]: connect 2.0
Connection length [replies/conn]: 1.000

Request rate: 10.1 req/s (99.1 ms/req)
Request size [B]: 70.0

Reply rate [replies/s]: min 10.0 avg 10.0 max 10.0 stddev 0.0 (1 sampl
es)
Reply time [ms]: response 3.4 transfer 2.6
Reply size [B]: header 281.0 content 7562.0 footer 0.0 (total 7843.0)
Reply status: 1xx=0 2xx=100 3xx=0 4xx=0 5xx=0

CPU time [s]: user 0.72 system 8.75 (user 7.3% system 88.3% total 95.5
%)
Net I/O: 78.0 KB/s (0.6*10^6 bps)

Errors: total 0 client-timo 0 socket-timo 0 connrefused 0 connreset 0
Errors: fd-unavail 0 addrunavail 0 ftab-full 0 other 0
```

Figura a

```

[root@server ~]# httpperf --server=www.serverapache.com --rate=10 --num
-conns=100
httpperf --client=0/1 --server=www.serverapache.com --port=80 --uri=/ -
-rate=10 --send-buffer=4096 --recv-buffer=16384 --num-conns=100 --num-
calls=1
Maximum connect burst length: 1

Total: connections 100 requests 100 replies 100 test-duration 9.909 s

Connection rate: 10.1 conn/s (99.1 ms/conn, <=1 concurrent connections
)
Connection time [ms]: min 5.1 avg 8.3 max 10.9 median 8.5 stddev 1.1
Connection time [ms]: connect 1.7
Connection length [replies/conn]: 1.000

Request rate: 10.1 req/s (99.1 ms/req)
Request size [B]: 73.0

Reply rate [replies/s]: min 10.0 avg 10.0 max 10.0 stddev 0.0 (1 sampl
es)
Reply time [ms]: response 4.2 transfer 2.4
Reply size [B]: header 265.0 content 7368.0 footer 0.0 (total 7633.0)
Reply status: 1xx=0 2xx=100 3xx=0 4xx=0 5xx=0

CPU time [s]: user 0.71 system 8.75 (user 7.1% system 88.3% total 95.5
%)
Net I/O: 75.9 KB/s (0.6*10^6 bps)

Errors: total 0 client-timo 0 socket-timo 0 connrefused 0 connreset 0
Errors: fd-unavail 0 addrunavail 0 ftab-full 0 other 0
[root@server ~]#

```

Figura b

Figura IV.58. Porcentaje de uso de CPU con prueba sin carga cliente-servidor para **a.**

Servidor IIS y b. Servidor Apache

```

[root@server ~]# httpperf --server=www.serveriis.com --rate=10 --num-conn=5000
httpperf --client=0/1 --server=www.serveriis.com --port=80 --uri=/ --rate=10 --send-buffer=4096 --recv-buffer=16384 --num-conn=5000 --num-connections=1
Maximum connect burst length: 16

Total: connections 5000 requests 5000 replies 4974 test-duration 499.964 s

Connection rate: 10.0 conn/s (100.0 ms/conn, <=276 concurrent connections)
Connection time [ms]: min 0.9 avg 896.8 max 45022.2 median 4.5 stddev 4504.3
Connection time [ms]: connect 1019.9
Connection length [replies/conn]: 1.000

Request rate: 10.0 req/s (100.0 ms/req)

Request size [B]: 70.0

Reply rate [replies/s]: min 0.0 avg 9.9 max 40.2 stddev 4.0 (99 samples)
Reply time [ms]: response 5.9 transfer 2.0
Reply size [B]: header 281.0 content 7562.0 footer 0.0 (total 7843.0)
Reply status: 1xx=0 2xx=4974 3xx=0 4xx=0 5xx=0

CPU time [s]: user 12.99 system 482.15 (user 2.6% system 96.4% total 99.0%)
Net I/O: 76.9 KB/s (0.6*10^6 bps)

Errors: total 26 client-timo 0 socket-timo 0 connrefused 0 connreset 26
Errors: fd-unavail 0 addrunavail 0 ftab-full 0 other 0
[root@server ~]# _

```

Figura a

```

[root@server ~]# httpperf --server=www.serverapache.com --rate=10 --num-
m-conns=5000
httpperf --client=0/1 --server=www.serverapache.com --port=80 --uri=/
--rate=10 --send-buffer=4096 --recv-buffer=16384 --num-conns=5000 --n
um-calls=1
Maximum connect burst length: 2

Total: connections 5000 requests 5000 replies 5000 test-duration 499.
904 s

Connection rate: 10.0 conn/s (100.0 ms/conn, <=3 concurrent connectio
ns)
Connection time [ms]: min 1.9 avg 5.1 max 248.3 median 3.5 stddev 7.8
Connection time [ms]: connect 1.5
Connection length [replies/conn]: 1.000

Request rate: 10.0 req/s (100.0 ms/req)
Request size [B]: 73.0

Reply rate [replies/s]: min 10.0 avg 10.0 max 10.0 stddev 0.0 (99 sam
ples)
Reply time [ms]: response 2.5 transfer 1.1
Reply size [B]: header 265.0 content 7368.0 footer 0.0 (total 7633.0)
Reply status: 1xx=0 2xx=5000 3xx=0 4xx=0 5xx=0

CPU time [s]: user 9.30 system 487.30 (user 1.9% system 97.5% total 9
9.3%)
Net I/O: 75.3 KB/s (0.6*10^6 bps)

Errors: total 0 client-timo 0 socket-timo 0 connrefused 0 connreset 0
Errors: fd-unavail 0 addrunavail 0 ftab-full 0 other 0
[root@server ~]# █

```

Figura b

Figura IV.59. Porcentaje de uso de CPU, prueba con carga cliente-servidor para **a.**

Servidor IIS y **b.** Servidor Apache

4.4 ANÁLISIS DE RESULTADOS

La necesidad de valorar el desempeño de cada uno de los servidores después de las pruebas realizadas y el análisis de los parámetros establecidos para la medición conlleva a la asignación de pesos, para ello, se establece un valor numérico de acuerdo a las siguientes necesidades:

Tabla IV.VII. Descripción de Asignación de pesos

<i>Pesos</i>	<i>Referencia</i>
<i>1</i>	Si el resultado de la medición de la variable de un servidor genera ventaja sobre el otro servidor
<i>0</i>	Si el resultado de la medición de la variable de un servidor genera desventaja en relación con el otro servidor

Elaborado por: Autora de Tesis

Se considera una desventaja cuando la diferencia en los resultados tiene un valor significativo y el mismo representa una variación.

Tabla IV.VIII.Resultados Prueba 1.Estrés, 10 usuarios 1000 peticiones por usuario

	Apache		IIS	
Análisis de Tráfico				
Trafico Total	75,567 Mb	151,281 paquetes	76,176 Mb	143,290 paquetes
Trafico Broadcast Enviado	11,714 Mb	97 paquetes	10,279 Mb	84 paquetes
Tráfico Multicast Enviado	8,938 Mb	143 paquetes	8,938 Mb	143 paquetes
Análisis IP				
IP	75,565 Mb	146,555 paquetes	75,891 Mb	139,538 paquetes
Análisis TCP				
Paquetes SYN tcp	1,268 Mb	20,148 paquetes	1,248 Mb	19,822 paquetes

Paquetes FIN tcp	8,005 Mb	20,150 paquetes	7,894 Mb	19,834 paquetes
Conexiones inicializadas	10,075 conexiones		9,917 conexiones	
Conexiones establecidas	10,075 conexiones		9,917 conexiones	
Análisis HTTP				
Conexiones HTTP	10,073 conexiones		9,910 conexiones	
Peticiones HTTP	10,073 conexiones		9,961 conexiones	
Análisis DNS				
Peticiones DNS	308		277	
Respuestas DNS	21		15	
Errores	24		28	
Análisis NetBios				

Netbios TCP	3,078	21	3,078	21
Netbios UDP	8,319	74	8,221	68
<i>Análisis ARP</i>				
Peticiones	1,000 Kb	16	832 B	13
Respuestas	832 B	13	448 B	7
<i>Tiempo total de Respuesta</i>				
Tiempo de Respuesta (ping ICMP)	79 ms		166 ms	

Elaborado por: Autora de Tesis

Tabla IV.IX. Asignación de pesos para obtener el servidor apropiado a nuestra necesidad

Prueba 1

	APACHE	IIS	Explicación
Tráfico Total	0	1	IIS genera menos trafico total que Apache
Trafico Broadcast Enviado	0	1	
Tráfico Multicast Enviado	1	1	
IP	0	1	IIS genera menos paquetes IP que Apache
Paquetes SYN TCP	0	1	IIS genera menos paquetes TCP que Apache
Paquetes FIN TCP	0	1	
Conexiones inicializadas TCP	0	1	
Conexiones establecidas TCP	0	1	
Conexiones HTTP	0	1	IIS genera menos paquetes

Peticiones HTTP	0	1	HTTP que Apache
Peticiones DNS	0	1	IIS genera menos peticiones DNS, apache genera menos errores DNS
Respuestas DNS	0	1	
Errores	1	0	
Netbios TCP	1	1	IIS y Apache genera igual numero de paquetes netbios de conexión TCP, IIS genera menos paquetes netbios de conexión UDP
Netbios UDP	0	1	
Peticiones ARP	1	1	Los dos servidores general similar numero de paquetes ARP
Respuestas ARP	1	1	
Tiempo de Respuesta Total	1	0	Apache responde más rápido a una solicitud mientras genera tráfico
Medición de variables favorables	6	16	

Elaborado por: Autora de Tesis

18 variables → 100%

6 x = 33.33%

16 x = 88.88%

Resultado de esta prueba: Apache 33.33 %, IIS 88,88%

Tabla IV.X.Resultados Prueba2. Carga Constante por 2 minutos con 10 usuarios simultáneos

	Apache	IIS
Análisis de Ancho de Banda t= 80s		
Cliente	3000 kbits/s	3400 kbits/s
Servidor	100 kbits/s	125 kbits/s
Análisis de Eficiencia t=80s		
Trafico de Red	95 kbits/s	105 kbits/s
Memoria del sistema disponible	2,047 Mb	2,047 Mb
Carga CPU Local	35%	20%

Elaborado por: Autora de Tesis

Tabla IV.XI.Resultados Prueba 2

	APACHE	IIS	Explicación
Ancho de Banda del Cliente	1	0	Cliente en prueba utiliza menor ancho de banda utilizando el servidor Apache
Ancho de Banda del Servidor	1	0	Servidor bajo prueba utiliza menor ancho de banda utilizando el servidor Apache
Trafico de Red	1	0	Trafico de red generado es menor en Apache
Memoria del sistema disponible	1	1	Ambos servidores ocupan el mismo espacio de memoria disponible
Carga CPU Local	0	1	Cliente genera menos carga de CPU cuando realiza la prueba a IIS
Medición de variables favorables	4	2	

Elaborado por: Autora de Tesis

5 variables → 100%

4 x = 80%

2 x = 40%

Resultado de esta prueba: Apache 80 %, IIS 40%

Tabla IV.XII. Resultados Prueba3.Carga Variable por 2 minutos con 10 usuarios simultáneos

	Apache	IIS
Análisis de Ancho de Banda t= 80s		
Cliente	3600 kbits/s	3400 kbits/s
Servidor	3200 kbits/s	4400 kbits/s
Análisis de Eficiencia t=80s		
Trafico de Red	80 Kbits/s	95 kbits/s
Memoria del sistema disponible	2,047 Mb	2,047 Mb
Carga CPU Local	45 %	20 %

Elaborado por: Autora de Tesis

Tabla IV.XIII. Resultados Prueba 3

	APACHE	IIS	Explicación
Ancho de Banda del Cliente	0	1	Cliente en prueba utiliza menor ancho de banda utilizando el servidor IIS
Ancho de Banda del Servidor	1	0	Servidor bajo prueba utiliza menor ancho de banda utilizando el servidor IIS
Trafico de Red	1	0	Trafico de red generado es menor en Apache
Memoria del sistema disponible	1	1	Ambos servidores ocupan el mismo espacio de memoria disponible
Carga CPU Local	0	1	Cliente genera menos carga de CPU cuando realiza la prueba a IIS
Medición de variables favorables	3	3	

Elaborado por: Autora de Tesis

5 variables → 100%

3 x = 60%

3 x = 60%

Resultado de esta prueba: Apache 60 %, IIS 60%

Tabla IV.XIV. Resultados Prueba 4.Latencia usando ping IP, httping

	Apache		IIS	
Latencia	Sin Carga 22.84 ms	Con Carga 46.74 ms	Sin Carga 33.38 ms	Con Carga 57.88 ms
Velocidad de transferencia	Sin Carga 58775 KB	Con Carga 2192 KB	Sin Carga 54711 KB	Con Carga 19505 KB

Elaborado por: Autora de Tesis

Tabla IV.XV. Resultados Prueba 4

	APACHE		IIS		Explicación
	Sin carga	Con carga	Sin carga	Con carga	
Latencia (ms)	1	1	0	0	Apache sometido o no a carga tiene una velocidad de respuesta más rápida que IIS
Velocidad de transferencia (KB)	0	1	1	0	En cambio IIS sin carga tiene mayor velocidad de transferencia pero apache con carga deja ver sus ventajas
Medición de variables favorables	1	2	1	0	<i>“Apache e IIS sin carga poseen las mismas características de respuesta, Apache con carga es superior a IIS”</i>

Elaborado por: Autora de Tesis

4 variables → 100%

3 x = 75%

1 x = 25%

Resultado de esta prueba: Apache 75 %, IIS 25%

Tabla IV.XVI. Resultados Prueba 5.Sobre carga de peticiones

	Apache		IIS	
Tiempo de ida y vuelta promedio	Sin Carga 13,6 ms	Con Carga 16,2 ms	Sin Carga 12,3 ms	Con Carga 17,3 ms
Peticiones por segundo	Sin Carga 171,85	Con Carga 32,011	Sin Carga 234,62	Con Carga 25,058
Tiempo por petición	Sin Carga 29,095 ms	Con Carga 32,011 ms	Sin Carga 21,311 ms	Con Carga 25,058 ms

Elaborado por: Autora de Tesis

Tabla IV.XVII. Resultados Prueba 5

	Apache		IIS		Explicación
	Sin carga	Con carga	Sin carga	Con carga	
Tiempo de ida y vuelta promedio	0	1	1	0	El tiempo promedio de ida y vuelta para IIS sin carga
Peticiones por segundo que soporta	0	1	1	0	Apache soporta mayor número de peticiones con carga, IIS soporta mayor número de peticiones sin carga
Tiempo en atender a cada petición	0	1	1	0	Apache atiende más rápido mayor número de peticiones con carga, IIS atiende más rápido mayor número de peticiones sin carga
Medición de variables favorables	0	3	3	0	<i>“Apache tiene un mayor desempeño cuando está con carga de peticiones”</i>

Elaborado por: Autora de Tesis

6 variables → 100%

3 x = 50%

3 x = 50%

Resultado de esta prueba: Apache 50 %, IIS 50%

Tabla IV.XVIII. Resultados Prueba 6. Sobre carga de usuarios

	Apache	IIS
Tiempo de ida y vuelta promedio	4,887 ms	--
Peticiones por segundo	206,41	--
Tiempo por petición	2443,640 ms	--

Elaborado por: Autora de Tesis

Tabla IV.XIX. Resultados Prueba 6

	Apache	IIS	Explicación
Tiempo de ida y vuelta promedio	1	0	Apache supero la prueba
Peticiones por segundo que soporta	1	0	
Tiempo en atender a cada petición	1	0	
Medición de variables favorables	3	0	

Elaborado por: Autora de Tesis

3 variables → 100%

3 x = 100%

0 x = 0%

Resultado de esta prueba: Apache 100 %, IIS 0%

Tabla IV.XX. Resultados Prueba 7. Porcentaje de uso del CPU

	Apache		IIS	
Uso del CPU	Sin Carga 7,1%	Con Carga 97,5%	Sin Carga 7,3%	Con Carga 96,4%
Total de Errores	0		26	

Elaborado por: Autora de Tesis

Tabla IV.XXI. Resultados Prueba 7

	Apache		IIS		Explicación
	Sin carga	Con carga	Sin carga	Con carga	
Uso del CPU	1	0	1	1	Apache
Total de Errores	1		0		Apache n tubo ningún error
Medición de variables favorables	2		2		

Elaborado por: Autora de Tesis

3 variables → 100%

$$2 \quad x \quad = \quad 66.66\%$$

$$2 \quad x \quad = \quad 66.66\%$$

Resultado de esta prueba: Apache 66.66 %, IIS 33.33%

Tabla IV.XXII. Resultados de las Pruebas

	Apache	IIS
	%	%
Prueba 1	33.33	88.88
Prueba 2	80	40
Prueba 3	60	60
Prueba 4	75	25
Prueba 5	50	50
Prueba 6	100	0
Prueba 7	66.66	66.66
Promedio	66.43%	47.22%
Pruebas		

Elaborado por: Autora de Tesis

$$100 \% \rightarrow 80\%$$

$$66.43\% \quad x \quad = \quad 53.14 \%$$

$$47.22\% \quad x \quad = \quad 37.78\%$$

Una vez calculada la equivalencia se realiza la suma de los valores que se obtuvieron en: Característica de Servidor (20%), Rendimiento del servidor y Recursos del Sistema (80%) resultado que puede llegar al 100% de la comparativa.

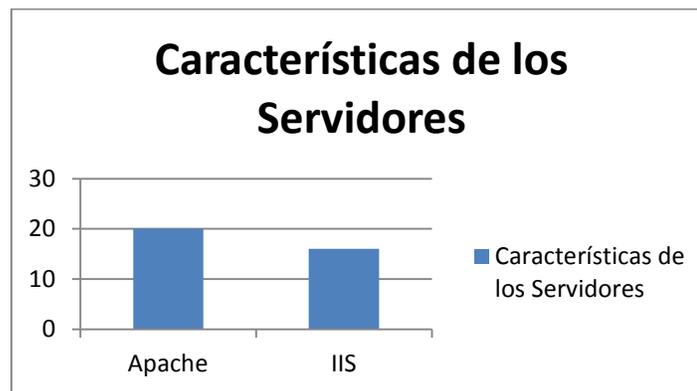


Figura IV.60. Grafica porcentual de las características de los servidores

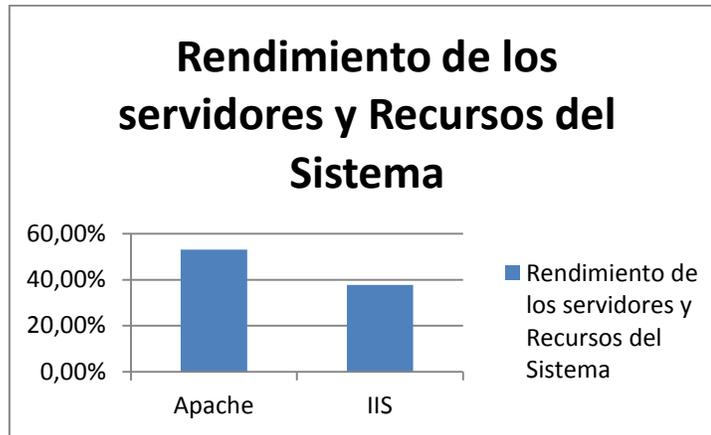


Figura IV.61. Grafica porcentual del rendimiento de los servidores y recursos del Sistema.

Tabla IV.XXIII. Resultado Final

	APACHE	IIS
	%	%
Característica de Servidor	20,00	16,00
Rendimiento del servidor y Recursos del Sistema	53.14	37.78
TOTAL	73.14%	53.78%

Elaborado por: Autora de Tesis

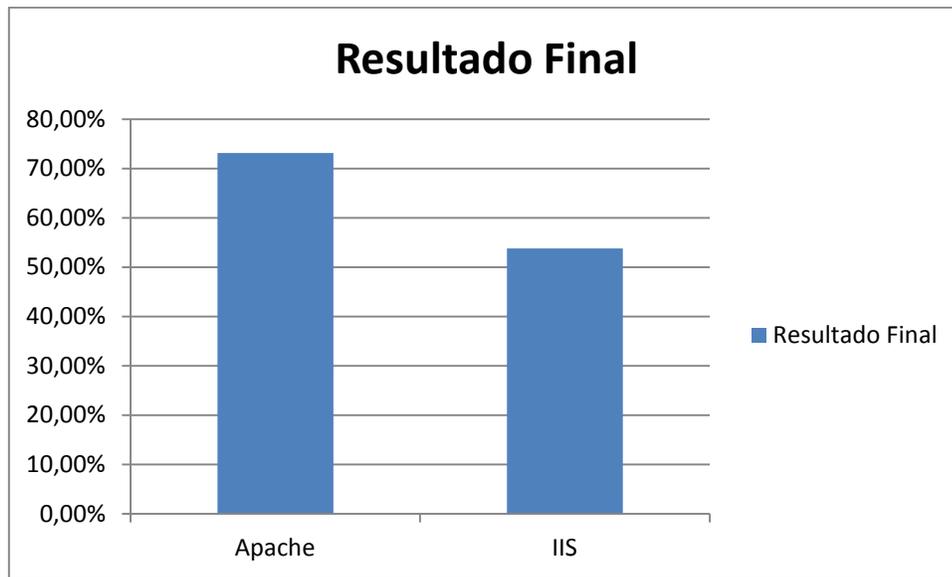


Figura IV.62. Grafica porcentual del Resultado Final de la Comparativa

Siendo el Servidor Apache con un 73.14% obtenido en la comparación quien supera a IIS que alcanzó un 53.78%.

4.5 RESULTADO FINAL DE LA COMPARATIVA

Una vez analizado cada servidor, el rendimiento y el uso de los recursos del sistema por cada uno se llega a la conclusión que en aquellas variables medidas de las 7 pruebas realizadas, en donde existieron diferencias significativas en sus valores, Apache supero a IIS en varias de ellas y en las que no existió diferencias significativas los dos servidores presentan una similar funcionalidad totalizando Apache 73.14% e IIS 53.78%.

CONCLUSIONES

- Mediante el uso de herramientas como Colasoft, Webserver Stress Tool y Comando de Shell de Linux, se logró analizar las características de cada uno de los servidores sometiéndoles a pruebas que se presentan en la vida real como carga de usuarios y peticiones.
- De las pruebas realizadas en esta investigación, se obtuvieron los siguientes resultados: acerca de las características Apache obtuvo el 20% por otra parte IIS obtuvo un 16%, con respecto al rendimiento del servidor Apache obtuvo el 53.14% e IIS un 37.78% dándonos como resultado final los siguientes valores: 73.14% en el uso de recursos y rendimiento del servidor apache frente a un 53.78% del servidor IIS.
- Apache se muestra como una alternativa real, fiable, escalable, segura y veloz frente a IIS.
- Al utilizar Apache se garantiza que una aplicación Web funcione más rápido sobre todo cuando existe cargas altas de peticiones y usuarios.
- Una desventaja que presenta apache es su difícil administración ya que su ambiente no es gráfico frente a IIS que usa un ambiente más amigable de administración.

RECOMENDACIONES

- Implementar esta investigación aumentando el número de páginas web que se hospedan en los servidores, con el fin de aumentar más el número de peticiones tanto por usuario como por página visitada.
- Realizar el ambiente de pruebas no como red local sino accediendo desde una red pública, con el fin de estudiar el cambio en los tiempos de acceso y atención a los clientes.
- Añadir la programación necesaria para acoplar una base de datos que forme parte de las variables analizadas por ejemplo el tiempo de respuesta a una petición de un registro.

RESUMEN

El estudio comparativo del rendimiento de Servidores Web de Virtualización determinó el mejor Servidor con el objetivo de escoger el más adecuado de acuerdo a su desempeño en el uso de recursos y rendimiento en un ambiente de virtualización en la ESPOCH.

El método apropiado para el estudio de comparación es el Analítico, que permite presentar información cualitativa (características técnicas) y cuantitativa (resultados de la pruebas).

En la investigación se usó los siguientes programas, Colasoft Capsa para monitorear el Tráfico de Red que se genera cuando uno o varios clientes realiza peticiones a los servidores mencionados, además, se configuró el software WebStress Tools, para simular la carga de usuarios y el número de peticiones que generan, adicional a ello, se usó los comandos Linux ab, httpperf y httping.

Se obtuvo los siguientes resultados de las pruebas realizadas: Apache 73.14% e IIS 53.78% demostrándose que Apache es la mejor opción en servidores Web, ya que permite una mejor utilización de recursos y un rendimiento más eficiente cuando está sometido a una mayor carga de usuarios.

Como conclusión el servidor Apache es superior a IIS en uso de sus Recursos y Rendimiento, ya que estos aspectos son claves al momento de elegir el Sistema de una empresa que puede trabajar 24 horas los 365 días,

Se recomienda su uso para implementar Máquinas Virtuales con características básicas (256Mb de RAM y 32Mb de Espacio en Disco), ofreciendo así la posibilidad de crear un servidor de servidores usando la técnica de virtualización.

SUMMARY

The comparative performance of Web servers Virtualization determined the best Server in order to choose the right one according to their performance in resource use and performance in a virtualization environment in ESPOCH.

The appropriate method for the study of comparison is the analytical, allowing cualitativa present information (technical features) and quantitative (test results).

The investigation will use the following programs, Colasoft Capsa to monitor network traffic that is generated when one or more customers made requests to the servers listed also configured WebStress Tools software to simulate user load and the number of requests that generate, in addition to this, we used Linux commands ab, httpperf and httping. He obtained the following results of tests performed: Apache and IIS 73.14% 53.78% showing that Apache is the best choice in Web servers, allowing better use of resources and more efficient performance when subjected to increased user load .

In conclusion, the Apache server than IIS is using its resources and performance, as these aspects are key when choosing the system of a company that can work 24 hours 365 days

Recommended for use with Virtual Machines to implement basic characteristics (256Mb RAM and 32Mb of disk space), thus offering the possibility of creating a server using server virtualization technology.

BIBLIOGRAFÍA

INTERNET

1. APACHE

<http://httpd.apache.org/>

2011-06-26

<http://hackeruna.com/2011/11/24/hosts-virtuales-en-apache-2-ubuntu/>

2011-07-01

<http://www.dragonjar.org/vulnerabilidad-en-el-servidor-apache-20.xhtml>

2011-07-01

<http://www.tail-f.com.ar/servicios/herramienta-para-medir-la-latencia-de-un-servidor-web.html>

2012-01-25

2. COLASOFT

<http://colasoft-mac-scanner.softonic.com/>

2011-11-25

3.EJEMPLOS DE HERRAMIENTAS DE EDICIÓN WEB

<http://www.infor.uva.es/~jv>

[ia/pordocente/node22.html](http://www.infor.uva.es/~jv/ia/pordocente/node22.html)

2011-11-29

4. EL NAVEGADOR WEB, BROWSER

<http://www.infor.uva.es/~jvegas/cursos/buendia/pordocente/node19.html>

2011-10-25

5. EL PROTOCOLO HTTP

http://www.cibernetia.com/manuales/introduccion_aplicaciones_web/2_1_fundamentos_web.php

[fundamentos_web.php](http://www.cibernetia.com/manuales/introduccion_aplicaciones_web/2_1_fundamentos_web.php)

2011-12-30

<http://www.infor.uva.es/~jvegas/cursos/buendia/pordocente/node20.html>

2011-12-30

6. FUNDAMENTOS DE LA WEB

http://www.cibernetia.com/manuales/introduccion_aplicaciones_web/2_1_fundamentos_web.php

[fundamentos_web.php](http://www.cibernetia.com/manuales/introduccion_aplicaciones_web/2_1_fundamentos_web.php)

2012-11-10

7. IIS

http://es.wikipedia.org/wiki/Internet_Information_Services

2011-06-25

<http://technet.microsoft.com/es-es/lit>

[}v=ws.10%29.aspx](#)

2011-06-26

<http://es.scribd.com/doc/27519905/8/Caracteristicas-de-IIS>

2011-06-26

8. SERVIDORES WEB

<http://www.w3.org/Consortium/>

2011-11-11

http://www.ciens.ucv.ve/ciencias/servidores_web.htm

2011-11-25

http://es.wikipedia.org/wiki/Servidor_web

2011-12-15

9. VIRTUALIZACION

http://es.wikipedia.org/wiki/M%C3%A1quina_virtual

2011-07-10

http://www.elquille.info/sistema/maquinas_virtuales.htm

2011-08-10

10. WEBSERVER STRESS TOOL

<http://webserver-stress-tool.sc>

2011-08-19

<http://webserver-stress-tool.softonic.com/descargar>

2012-01-21

11. WINDOWS SERVER 2008

http://es.wikipedia.org/wiki/Windows_Server_2008

2011-08-20

<http://www.microsoft.com/latam/technet/windowsserver/longhorn/evaluate/whitepaper.msp>

2011-10-10

<http://www.microsoft.com/es-xl/servidores-nube/windows-server/default.aspx>

2011-11-01

12. WINDOWS SERVER 2003

http://es.wikipedia.org/wiki/Windows_Server_2003

2011-09-10

<http://support.microsoft.com/ph/3198/es-es>

2011-12-12

<http://www.microsoft.com/spain/windows>

<03/evaluation/overview/>

<default.aspx>

2011-12-28

ANEXOS

ANEXO 1

SUMARY LOG PRUEBA USANDO WEBSERVER STRESS TOOLS PARA IIS

** Test Logfile by Webserver Stress Tool 7.2.2.261 Trial Version **
© 1998-2009 Paessler AG, <http://www.paessler.com>

Test run on 17/02/2012 20:17:35

** Project and Scenario Comments, Operator **

Results of period #1 (from 1 sec to 6 sec):

Completed Clicks: 1 with 0 Errors (=0,00%)
Average Click Time for 1 Users: 33 ms
Successful clicks per Second: 0,20 (equals 717,07 Clicks per Hour)

Results of period #2 (from 6 sec to 11 sec):

Completed Clicks: 1 with 0 Errors (=0,00%)
Average Click Time for 1 Users: 11 ms
Successful clicks per Second: 0,20 (equals 716,17 Clicks per Hour)

Results of period #3 (from 11 sec to 16 sec):

Completed Clicks: 1 with 0 Errors (=0,00%)
Average Click Time for 1 Users: 11 ms
Successful clicks per Second: 0,20 (equals 716,04 Clicks per Hour)

Results of period #4 (from 16 sec to 21 sec):

Completed Clicks: 1 with 0 Errors (=0,00%)
Average Click Time for 1 Users: 12 ms
Successful clicks per Second: 0,20 (equals 715,22 Clicks per Hour)

Results of period #5 (from 21 sec to 26 sec):

Completed Clicks: 1 with 0 Errors (=0,00%)
Average Click Time for 1 Users: 10 ms

Successful clicks per Second: als 714,93 Clicks per Hour)

Results of period #6 (from 26 sec to 31 sec):

Completed Clicks: 1 with 0 Errors (=0,00%)
Average Click Time for 1 Users: 11 ms
Successful clicks per Second: 0,20 (equals 714,92 Clicks per Hour)

Results of period #7 (from 31 sec to 36 sec):

Completed Clicks: 1 with 0 Errors (=0,00%)
Average Click Time for 1 Users: 11 ms
Successful clicks per Second: 0,20 (equals 715,07 Clicks per Hour)

Results of period #8 (from 36 sec to 41 sec):

Completed Clicks: 1 with 0 Errors (=0,00%)
Average Click Time for 1 Users: 12 ms
Successful clicks per Second: 0,20 (equals 714,65 Clicks per Hour)

Results of period #9 (from 41 sec to 46 sec):

Completed Clicks: 1 with 0 Errors (=0,00%)
Average Click Time for 1 Users: 12 ms
Successful clicks per Second: 0,20 (equals 714,79 Clicks per Hour)

Results of period #10 (from 46 sec to 51 sec):

Completed Clicks: 1 with 0 Errors (=0,00%)
Average Click Time for 1 Users: 12 ms
Successful clicks per Second: 0,20 (equals 715,07 Clicks per Hour)

Results of period #11 (from 51 sec to 56 sec):

Completed Clicks: 1 with 0 Errors (=0,00%)
Average Click Time for 1 Users: 10 ms
Successful clicks per Second: 0,20 (equals 714,66 Clicks per Hour)

Results of period #12 (from 56 sec to 62 sec):

Completed Clicks: 1 with 0 Errors (=0,00%)
Average Click Time for 1 Users: 11 ms
Successful clicks per Second: 0,20 (equals 714,49 Clicks per Hour)

Results of period #13 (from 62 sec to 67 sec):

Completed Clicks: 1 with 0 Errors (=0,00%)
Average Click Time for 1 Users: 11 ms
Successful clicks per Second: 0,20 (equals 715,12 Clicks per Hour)

Results of period #14 (from 67 sec to 72 sec):

Completed Clicks: 1 with 0 Errors (%)
 Average Click Time for 1 Users:
 Successful clicks per Second: 0,20 (equals 714,94 Clicks per Hour)

Results of period #15 (from 72 sec to 77 sec):

 Completed Clicks: 1 with 0 Errors (=0,00%)
 Average Click Time for 1 Users: 11 ms
 Successful clicks per Second: 0,20 (equals 714,79 Clicks per Hour)

Results of period #16 (from 77 sec to 82 sec):

 Completed Clicks: 1 with 0 Errors (=0,00%)
 Average Click Time for 1 Users: 11 ms
 Successful clicks per Second: 0,20 (equals 715,08 Clicks per Hour)

Results of period #17 (from 82 sec to 87 sec):

 Completed Clicks: 1 with 0 Errors (=0,00%)
 Average Click Time for 1 Users: 13 ms
 Successful clicks per Second: 0,20 (equals 715,07 Clicks per Hour)

Results of period #18 (from 87 sec to 92 sec):

 Completed Clicks: 1 with 0 Errors (=0,00%)
 Average Click Time for 1 Users: 12 ms
 Successful clicks per Second: 0,20 (equals 715,07 Clicks per Hour)

Results of period #19 (from 92 sec to 97 sec):

 Completed Clicks: 1 with 0 Errors (=0,00%)
 Average Click Time for 1 Users: 10 ms
 Successful clicks per Second: 0,20 (equals 715,07 Clicks per Hour)

Results of period #20 (from 97 sec to 102 sec):

 Completed Clicks: 1 with 0 Errors (=0,00%)
 Average Click Time for 1 Users: 12 ms
 Successful clicks per Second: 0,20 (equals 715,06 Clicks per Hour)

Results of period #21 (from 102 sec to 107 sec):

 Completed Clicks: 1 with 0 Errors (=0,00%)
 Average Click Time for 1 Users: 11 ms
 Successful clicks per Second: 0,20 (equals 715,07 Clicks per Hour)

Results of period #22 (from 107 sec to 112 sec):

 Completed Clicks: 1 with 0 Errors (=0,00%)
 Average Click Time for 1 Users: 10 ms
 Successful clicks per Second: 0,20 (equals 714,93 Clicks per Hour)

```

Results of period #23 (from 1:          , 117 sec ):
*****:                               *****
Completed Clicks: 1 with 0 Errors      (0%)
Average Click Time for 1 Users:
Successful clicks per Second: 0,20 (equals 713,82 Clicks per Hour)

```

```

Results of complete test
*****

```

```

** Results per URL for complete test **

```

```

URL#1 (1): Average Click Time 12 ms, 23 Clicks, 0 Errors
URL#2 (2): 0 Clicks, 0 Errors
URL#3 (3): 0 Clicks, 0 Errors
URL#4 (4): 0 Clicks, 0 Errors

```

```

Total Number of Clicks: 23 (0 Errors)
Average Click Time of all URLs: 12 ms

```

```

!! Glossary:

```

```

!! Click: A simulated mouse click of a user sending a request (one of
the URLs from the URL list) to the server and immediately requesting
any necessary redirects, frames and images (if enabled).

```

```

!! Request: A HTTP request sent to the server regardless of an answer.

```

```

!! Hit: A completed HTTP request (i.e. sent to the server and answered
completely). Hits can be the PAGE request of a "click" or its frames,
images etc.

```

```

!! Time for DNS: Time to resolve a URL's domain name using the client
system's current DNS server.

```

```

!! Time to connect: Time to set up a connection to the server.

```

```

!! Time to first byte (TFB): Time between initiating a request and
receiving the first byte of data from the server.

```

```

!! Click Time: The time a user had to wait until his "click" was
finished (including redirections/frames/images etc.).

```

```

!! User Bandwidth: The bandwidth a user was able to achieve.

```

```

!! Sent Requests: Number of requests sent to the server during a
period.

```

```

!! Received Requests: Number of answers received from the server
during a period.

```

URLs to Test

URL#	Name	Click Delay [s]	URL	POST data (or @filename@)	Username	Password
1	1	5	http://www.serveriis.com/			
2	2	5	http://www.serveriis.com/Services.html			
3	3	5	http://www.serveriis.com/Products.html			
4	4	5	http://www.serveriis.com/Contacts.html			

Results per User

User No.	Clicks	Hits	Errors	Avg. Click Time [ms]	Bytes	kbit/s	Cookies
1	25	24	0	12	188.160	5.088,38	

Results per URL

URL No.	Name	Clicks	Errors	Errors [%]	Time Spent [ms]	Avg. Click Time [ms]	
1		1	23	0	0,00	282	12
2		2	0	0		0	
3		3	0	0		0	
4		4	0	0		0	

ANEXO 2

SUMARY LOG PRUEBA USANDO WEBSERVER STRESS TOOLS PARA APACHE

** Test Logfile by Webserver Stress Tool 7.2.2.261 Trial Version **
© 1998-2009 Paessler AG, <http://www.paessler.com>

Test run on 17/02/2012 20:12:39

** Project and Scenario Comments, Operator **

Results of period #1 (from 0 sec to 5 sec):

Completed Clicks: 1 with 0 Errors (=0,00%)
Average Click Time for 1 Users: 15 ms
Successful clicks per Second: 0,20 (equals 717,01 Clicks per Hour)

Results of period #2 (from 5 sec to 11 sec):

Completed Clicks: 1 with 0 Errors (=0,00%)
Average Click Time for 1 Users: 13 ms
Successful clicks per Second: 0,20 (equals 708,24 Clicks per Hour)

Results of period #3 (from 11 sec to 16 sec):

Completed Clicks: 1 with 0 Errors (=0,00%)
Average Click Time for 1 Users: 23 ms
Successful clicks per Second: 0,20 (equals 716,36 Clicks per Hour)

Results of period #4 (from 16 sec to 21 sec):

Completed Clicks: 1 with 0 Errors (=0,00%)
Average Click Time for 1 Users: 14 ms
Successful clicks per Second: 0,20 (equals 716,31 Clicks per Hour)

Results of period #5 (from 21 sec to 26 sec):

Completed Clicks: 1 with 0 Errors (=0,00%)
Average Click Time for 1 Users: 14 ms
Successful clicks per Second: 0,20 (equals 716,22 Clicks per Hour)

Results of period #6 (from 26 sec to 31 sec):

Completed Clicks: 1 with 0 Errors (=0,00%)
Average Click Time for 1 Users: 14 ms
Successful clicks per Second: 0,20 (equals 712,48 Clicks per Hour)

Results of period #7 (from 31 sec to 36 sec):

```
*****
Completed Clicks: 1 with 0 Errors (=0,00%)
Average Click Time for 1 Users: 21 ms
Successful clicks per Second: 0,20 (equals 713,70 Clicks per Hour)

Results of period #8 (from 36 sec to 41 sec ):
*****
Completed Clicks: 1 with 0 Errors (=0,00%)
Average Click Time for 1 Users: 15 ms
Successful clicks per Second: 0,20 (equals 709,70 Clicks per Hour)

Results of period #9 (from 41 sec to 46 sec ):
*****
Completed Clicks: 1 with 0 Errors (=0,00%)
Average Click Time for 1 Users: 14 ms
Successful clicks per Second: 0,20 (equals 711,21 Clicks per Hour)

Results of period #10 (from 46 sec to 51 sec ):
*****
Completed Clicks: 1 with 0 Errors (=0,00%)
Average Click Time for 1 Users: 14 ms
Successful clicks per Second: 0,20 (equals 710,62 Clicks per Hour)

Results of period #11 (from 51 sec to 56 sec ):
*****
Completed Clicks: 1 with 0 Errors (=0,00%)
Average Click Time for 1 Users: 19 ms
Successful clicks per Second: 0,20 (equals 707,70 Clicks per Hour)

Results of period #12 (from 56 sec to 61 sec ):
*****
Completed Clicks: 1 with 0 Errors (=0,00%)
Average Click Time for 1 Users: 16 ms
Successful clicks per Second: 0,20 (equals 716,18 Clicks per Hour)

Results of period #13 (from 61 sec to 66 sec ):
*****
Completed Clicks: 1 with 0 Errors (=0,00%)
Average Click Time for 1 Users: 13 ms
Successful clicks per Second: 0,20 (equals 715,90 Clicks per Hour)

Results of period #14 (from 66 sec to 71 sec ):
*****
Completed Clicks: 1 with 0 Errors (=0,00%)
Average Click Time for 1 Users: 15 ms
Successful clicks per Second: 0,20 (equals 716,04 Clicks per Hour)

Results of period #15 (from 71 sec to 76 sec ):
*****
Completed Clicks: 1 with 0 Errors (=0,00%)
Average Click Time for 1 Users: 15 ms
Successful clicks per Second: 0,20 (equals 714,93 Clicks per Hour)
```

Results of period #16 (from 76 sec to 81 sec):

Completed Clicks: 1 with 0 Errors (=0,00%)
Average Click Time for 1 Users: 11 ms
Successful clicks per Second: 0,20 (equals 716,32 Clicks per Hour)

Results of period #17 (from 81 sec to 86 sec):

Completed Clicks: 1 with 0 Errors (=0,00%)
Average Click Time for 1 Users: 12 ms
Successful clicks per Second: 0,20 (equals 715,90 Clicks per Hour)

Results of period #18 (from 86 sec to 91 sec):

Completed Clicks: 1 with 0 Errors (=0,00%)
Average Click Time for 1 Users: 11 ms
Successful clicks per Second: 0,20 (equals 716,05 Clicks per Hour)

Results of period #19 (from 91 sec to 96 sec):

Completed Clicks: 1 with 0 Errors (=0,00%)
Average Click Time for 1 Users: 12 ms
Successful clicks per Second: 0,20 (equals 716,18 Clicks per Hour)

Results of period #20 (from 96 sec to 101 sec):

Completed Clicks: 1 with 0 Errors (=0,00%)
Average Click Time for 1 Users: 12 ms
Successful clicks per Second: 0,20 (equals 703,97 Clicks per Hour)

Results of period #21 (from 101 sec to 106 sec):

Completed Clicks: 1 with 0 Errors (=0,00%)
Average Click Time for 1 Users: 11 ms
Successful clicks per Second: 0,20 (equals 715,39 Clicks per Hour)

Results of period #22 (from 106 sec to 111 sec):

Completed Clicks: 1 with 0 Errors (=0,00%)
Average Click Time for 1 Users: 12 ms
Successful clicks per Second: 0,20 (equals 715,88 Clicks per Hour)

Results of period #23 (from 111 sec to 116 sec):

Completed Clicks: 1 with 0 Errors (=0,00%)
Average Click Time for 1 Users: 11 ms
Successful clicks per Second: 0,20 (equals 714,96 Clicks per Hour)

Results of complete test

** Results per URL for complete test **

URL#1 (1): Average Click Time 14 ms, 23 Clicks, 0 Errors

URL#2 (2): 0 Clicks, 0 Errors

URL#3 (3): 0 Clicks, 0 Errors

URL#4 (4): 0 Clicks, 0 Errors

Total Number of Clicks: 23 (0 Errors)

Average Click Time of all URLs: 14 ms

!! Glossary:

!! Click: A simulated mouse click of a user sending a request (one of the URLs from the URL list) to the server and immediately requesting any necessary redirects, frames and images (if enabled).

!! Request: A HTTP request sent to the server regardless of an answer.

!! Hit: A completed HTTP request (i.e. sent to the server and answered completely). Hits can be the PAGE request of a "click" or its frames, images etc.

!! Time for DNS: Time to resolve a URL's domain name using the client system's current DNS server.

!! Time to connect: Time to set up a connection to the server.

!! Time to first byte (TFB): Time between initiating a request and receiving the first byte of data from the server.

!! Click Time: The time a user had to wait until his "click" was finished (including redirections/frames/images etc.).

!! User Bandwidth: The bandwidth a user was able to achieve.

!! Sent Requests: Number of requests sent to the server during a period.

!! Received Requests: Number of answers received from the server during a period.

URLs to Test

URL#	Name	Click Delay [s]	URL	POST data (or @filename@)	Username	Password
1	1	5	http://www.serverapache.com/			
2	2	5	http://www.serverapache.com/Services.html			
3	3	5	http://www.serverapache.com/Products.html			
4	4	5	http://www.serverapache.com/Contacts.html			

Results per User

User No.	Clicks	Hits	Errors	Avg. Click Time [ms]	Bytes	kbit/s	Cookies
1	25	24	0	14	182.712	4.295,67	

Results per URL

URL No.	Name	Clicks	Errors	Errors [%]	Time Spent [ms]	Avg. Click Time [ms]	
1		1	23	0	0,00	324	14
2		2	0	0		0	
3		3	0	0		0	
4		4	0	0		0	