THE POSSIBILITIES ARE INFINITE FUJITSU

WHITE PAPER

Version 1.1
September 2007

Benchmark overview SPECweb2005

Pages 7

Abstract

This document describes the web server benchmark SPECweb2005 which was developed by the Standard Performance Evaluation Corporation (SPEC) and issued in 2005. This new version continues the series of SPEC benchmarks for assessing web servers under objective and reality loads. Some challenging extensions have been included in order to consider the wide range of new web technologies:

- Modelling real Internet users.
- Dynamic page contents either with PHP or JSP implementation.
- Transfer of page contents via two parallel HTTP connections.
- Three different user loads: Banking (HTTPS), Ecommerce (HTTP and HTTPS) and Support (HTTP). HTTPS stands for HTTP Secure.
- Simulation of browser caching effects by using If-Modified-Since requests (IMS).
- File accesses according to the patterns of current web server applications in real environments.
- Java-based implementation of the Benchmark control for clear and portable code.

The benchmark tests both the server hardware, in particular CPU, network cards, main memory and disk subsystem as well as the software components, such as web server, JVM (Java Virtual Machine) and operating system. Comparisons across all manufacturers are possible based on the results using standardized loads and the rules for implementing the SPECweb2005 benchmark as defined and tested by SPEC.

Contents

SPECweb2005 – an overview

In order to compare different computer systems the ideal situation would be to use the end-customer's application software for the test if suitable. However, as it is usually not possible to compare different systems in this way due to the amount of effort, lack of time or sheer complexity, standardized benchmarks have been developed for different application areas. These benchmarks enable an objective assessment of the entire system or its components based on defined tests, loads and rules for standardized execution which have all been tested by an independent group.

The "Standard Performance Evaluation Corporation" (SPEC) is an organization which looks at the development and publication of such benchmarks. The SPEC consortium includes representatives from the leading computer industry companies, including Fujitsu. One of the benchmarks issued by SPEC is SPECweb2005 which replaced the older versions in October 2005, i.e. SPECweb99 and SPECweb99_SSL. The aim of this document is to describe the benchmark principles, the execution of the measurements and the interpretation of the results.

SPECweb2005 is used to assess web servers. As today's web servers cover a very wide range of tasks, the benchmark uses three different loads in the following application areas: banking, e-commerce and support. The performance measurement selected was the number of users that can be simultaneously served on the SUT (system under test).

The three loads are based on real applications and contain the following tasks:

- **SPECweb2005_Banking** – Typical requests which would be sent by customers of an on-line bank were simulated, such as log-on/log-off, bank balance inquiry, money transfers, show and modify the user profile etc. The log-in includes setting up an SSL connection which is used for all further actions.
- **SPECweb2005_Ecommerce** – On-line trading with computers is simulated. The users can look at the pages, look at the products, compile and place orders. The first steps are via non-secure connections. SSL-encrypted connections are generated as soon as an order is to be placed and sent.
- **SPECweb2005_Support** – Support page requests are simulated. The users can look at the pages, look at lists of available products and download the appropriate files. All inquiries are not encrypted.

The three loads are measured consecutively with separate results. The overall metric SPECweb2005 is an average of the three individual results. The used metrics and their context are presented in detail at the end of this document.

The user behaviour is mapped in sessions. Threads are started according to the number of users which continually create requests, sending them to the web server. Waiting times (think times) exist between the requests; on average 10 seconds for Banking and Ecommerce and 5 seconds for Support. Banking and Ecommerce must have response times which meet certain criteria between sending requests and the complete receipt of the pages, while Support also has certain criteria for the throughput of received data per time unit. These criteria are referred to as Quality of Service (QoS) criteria. A content-related series of requests and answers is referred to as a session. In a thread a session is processed and then a new session is started and processed once the previous session has ended. This procedure simulates the behaviour of web page users who open a page, make http requests and then leave the page again.

SPECweb2005 is based on a page-oriented model, i.e. requests are sent to dynamic pages generated on the web server including embedded image files. The previous versions predominantly used static pages. Each load profile has a pool of pages from which - per request – a random distribution defined according to the benchmark specification is selected. As is now common practice with browsers, the embedded image files are transferred from the web server via a second, parallel HTTP connection per session.

Benchmark components

SPECweb2005 basically consists of four main components: the load generators (client systems), the control system (prime client), the web server and the back-end simulator (BeSim). The following figure shows this components and their interaction.

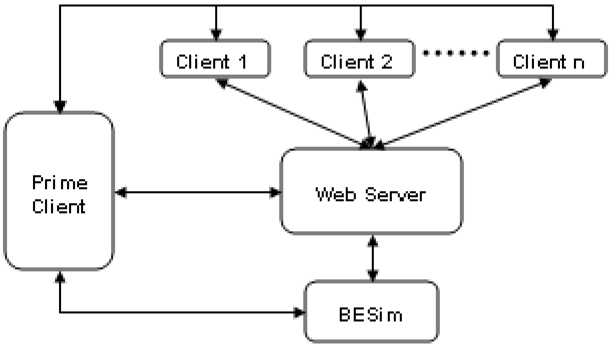


Figure 1: Logical components in SPECweb2005

Each component has the following tasks:

- A load generator program runs on the clients which builds the connections to the web server, sends page requests and receives answer pages. This program was written in Java for simple porting purposes. Several clients can exist on a physical system as they are logical components.
- The prime client initializes all the other systems, monitors the test, collects the results and evaluates them. As it is also a logical component, the prime client can be installed on a separate physical system or on one of the other client systems.
- The web server or SUT (System Under Test) comprises the hardware and software for processing the requests.
- The back-end simulator (BeSim) simulates the database and application parts of the application. The web server must connect up with the simulator in order to get information to process the HTTP requests completely, e.g. customer data for the dynamic creation of web pages.

The web server can consist of one single system or of several systems or nodes. The latter situation requires a load distribution component which accepts all HTTP requests and distributes them to the available web server nodes.

In addition to network cards for LAN connections shown in Figure 1 the web server requires a suitable storage configuration. The basis for the pages dynamically created by the web server is the files which are created using a tool provided by SPEC and which are stored in the file systems. This file pool has a load-independent, static part for each load profile that corresponds to the images embedded in the pages. However, the main point is that the number of files is proportional to the number of configured users, i.e. the size of the required file systems depends on the targeted performance.

SPECweb2005 test procedure

Before starting the benchmark, one or several client processes are started on each client system which listen on a network port. When all the client processes have been started, the client systems are ready for initialization via the prime client. The web service and application are started on the back-end simulator BeSim.

The prime client reads the test parameters from the configuration files and initializes the web server, BeSim and the other clients. The configuration files list the number of virtual users (sessions) to be created depending on the performance capability of the web server. The prime client forwards this information and additional information to the clients. When the initialization has been concluded, the prime client starts the benchmark run.

The prime client controls the various phases of a benchmark run which are illustrated in the following diagram:

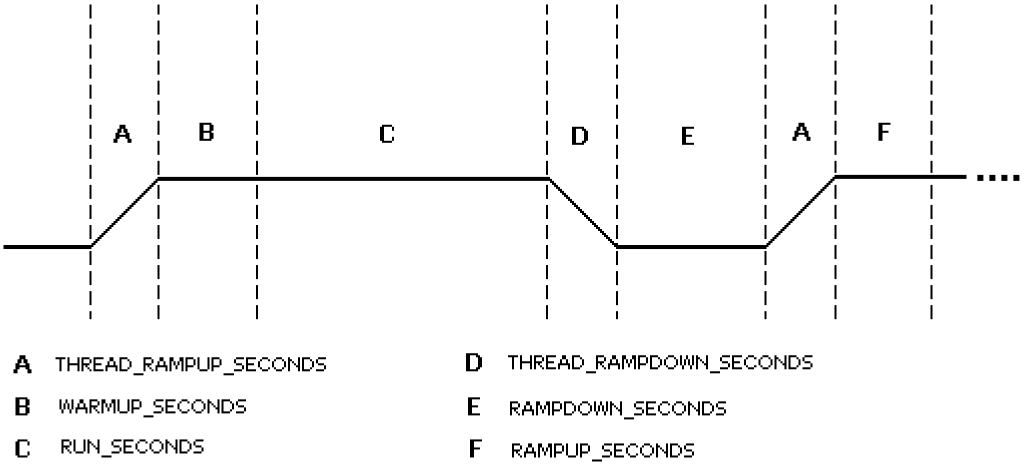


Figure 2: SPECweb2005 test phases

Each benchmark run begins with the ramp-up phase (A) which is at least 180 seconds but may last up to 600 seconds. This phase is aimed at starting the user sessions one after the other so that the entire load is not sent to the server in one go.

It is followed by the warm-up phase (B) during which the test system can prepare its caches before the actual measurement interval begins. At the end of this phase which lasts at least 300 and maximum 600 seconds, all the results recorded so far on the client systems are deleted, i.e. likewise all the errors which occurred before starting the measurement interval. They do not appear in the final result files.

The results of all the HTTP requests are recorded during the run phase (C) and the benchmark result is later calculated from the response times. This actual measurement interval lasts 1800 seconds.

In contrast to the ramp-up phase we now have the ramp-down phase (D) which is used to stop the created user sessions one after the other. The requests still sent during this time are no longer recorded and thus not taken into account in the final result.

There then follows an idle phase (E) of 300 seconds which gives the server and clients sufficient time to return to the non-load status.

The next run begins with the ramp-up phase (F) of 300 seconds which replaces phase (B) of the first run in the following runs. We can assume here that the caches have already been prepared via the previous test activities, thus reaching a stable status more quickly. Phase (B) typically lasts longer than phase (F) (cf. Figure 3).

At the end of each run the prime client collects the result data, compiles the data and writes the information in a result file. A total of three runs for each load profile are implemented. When all three runs have been ended, the report files are created in text format and in HTML format.

An attempt is made for each of the three load profiles to configure the largest possible number of users. The limit is defined by QoS criteria. The response time, until the page including the image files completely arrives, is recorded for each page requested and assessed as follows: GOOD if the response time is under two seconds; TOLERABLE if the response time is below four seconds.

One exception applies for the assessment criteria, namely downloading large files in the load profile Support. GOOD is when a transfer rate of at least 99,000 bytes per second is reached; TOLERABLE is 95,000 bytes per second.

The benchmark run is valid when at least 95% of the requests have been assessed as GOOD for each page and 99% of the requests as TOLERABLE.

Figure 3 shows an example of the SUT activity over a three-hour complete run of the load profile Support with 13604 configured sessions. The rhythm is clearly seen in the three measurement phases plus idle phases. The green (top) line means there was a count every ten seconds to see how many new requests were handled with GOOD – the blue (bottom) line shows those measured with only TOLERABLE. The third red curve of FAILS is almost identical to the x axis, i.e. hardly any FAILS occurred. The sum of all requests within a counting period of ten seconds is somewhat larger than the number of sessions as the average runtime for a request including waiting and response times is under ten seconds and on average slightly more than one request per session was handled.

The runs for the load profiles Banking and Ecommerce look like those in Figure 3. A complete measurement of the SPECweb2005 benchmark lasts about nine hours.

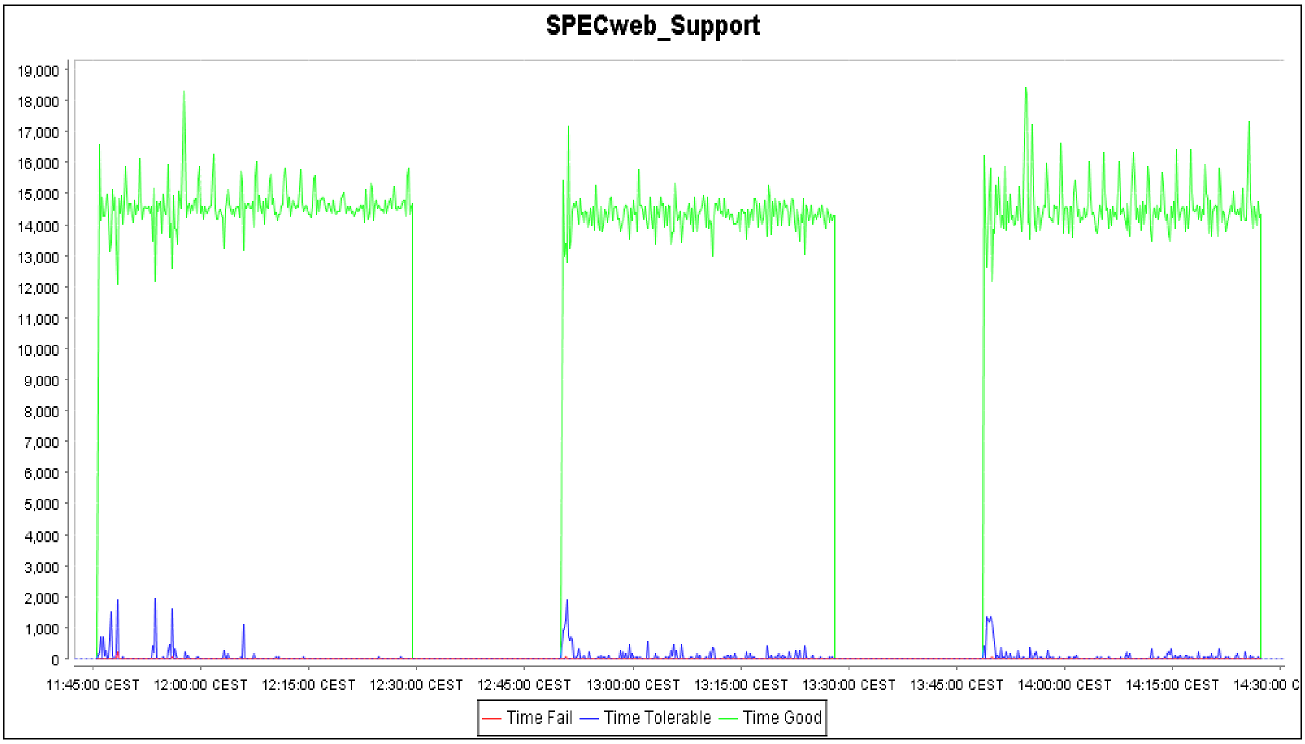


Figure 3: Three runs for the load profile Support and their QoS behaviour

Explanation of the test results

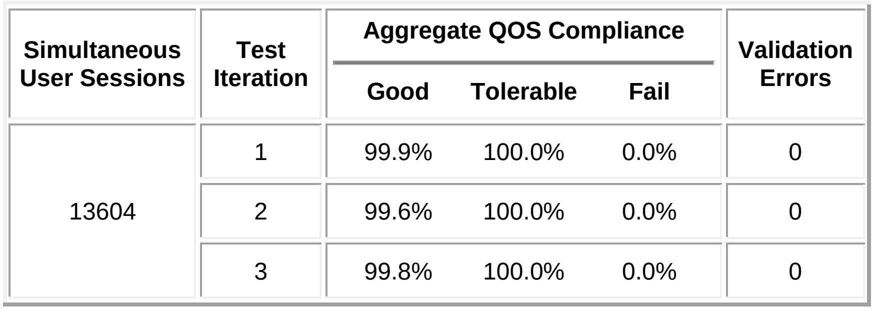
The primary performance metric SPECweb2005 states the factor at which the measured system is more powerful than a previously defined reference system whose performance is assumed to be 100. The reference system was an AMD Athlon 1.20 GHz based monoprocessor server with Linux operating system and Apache web server.

This primary metric SPECweb2005 is calculated from three load-specific secondary metrics as follows:

$$\text{SPECweb2005}=\sqrt[3]{\left(\frac{\text{bank\_measured}}{\text{bank\_reference}}\right)\times\left(\frac{\text{ecommerce\_measured}}{\text{ecommerce\_reference}}\right)\times\left(\frac{\text{support\_measured}}{\text{support\_reference}}\right)}\times 100$$

The individual results of the reference system are in the denominator of the quotients; the measured values of the SUT are in the numerator. The final SPECweb2005 result is the geometric average of the three quotients multiplied by 100. Among the various options of establishing an average value, the advantage of the geometric average is that the influence of the individual values on the average does not depend on their absolute size: this prevents an excellent individual result from dominating the final result.

The primary metric SPECweb2005 says nothing about the number of users supported by the SUT. However, the three secondary metrics are the number of users configured for the test as described above. This number of users is the same for all three runs per load profile. There are differences regarding the QoS criteria, i.e. in the percentage share of the three QoS classes GOOD, TOLERABLE and FAIL. In addition to the primary metric and the three secondary metrics, the certificate of a SPECweb2005 publication lists the QoS percentage shares for each of the three runs per load profile, as in figure 4 a certificate extract using the profile support as an example. This extract is taken from the same run as the graphic version in Figure 3.



Simultaneous User Sessions	Test Iteration	Aggregate QOS Compliance			Validation Errors
		Good	Tolerable	Fail	
13604	1	99.9%	100.0%	0.0%	0
	2	99.6%	100.0%	0.0%	0
	3	99.8%	100.0%	0.0%	0

Figure 4: Certificate extract for the load profile Support

The most significant influences on the performance of the SPECweb2005 benchmark are:

- the number of processors and their characteristics
- the memory subsystem
- the system bus
- the I/O subsystem, mostly used for networking, and for a minor part for storage I/O
- the capabilities of the operating system
- web server software
- the Java runtime environment (JVM)
- 64-bit address area support

SPECweb2005 is not suitable for sizing server configurations, but it offers excellent opportunities to compare web server products in the server environment. A competitive positioning in this benchmark means that the SUT with its hardware and software components is very suitable as a web server.

Literature

PRIMERGY Systems	http://ts.fujitsu.com/primergy
PRIMERGY Performance	http://ts.fujitsu.com/products/standard_servers/primergy_bov.html
SPECweb2005	http://www.spec.org/web2005

Contact

PRIMERGY Hardware

PRIMERGY Product Marketing
mailto:Primergy-PM@ts.fujitsu.com

PRIMERGY Performance and Benchmarks

PRIMERGY Performance and Benchmarks
mailto:primergy.benchmark@ts.fujitsu.com

Delivery subject to availability, specifications subject to change without notice, correction of errors and omissions excepted.
All conditions quoted (TCs) are recommended cost prices in EURO excl. VAT (unless stated otherwise in the text). All hardware and software names used are brand names and/or trademarks of their respective holders.

Published by department:

Enterprise Products
PRIMERGY Server
PRIMERGY Performance Lab
mailto:primergy.benchmark@ts.fujitsu.com

Internet:
http://ts.fujitsu.com/primergy

Extranet:
http://partners.ts.fujitsu.com/com/products/servers/primergy