WHITE PAPER

Version 1.1
January 2009

Comparing
Web Server Performance

Pages 10

Abstract

Four publications with the SPECweb2005 benchmark are available for the PRIMERGY TX150 S6. The results document the excellent suitability of the system for this area of application. Furthermore, the full potential of the benchmark is exhausted for the very first time: results with PHP for dynamic web pages are available alongside results with JSP, and *out-of-the-box* results are shown alongside highly optimized results. Rules of thumb follow for the different resource consumption of PHP and JSP, as well as with regard to the scope for investment in sophisticated software tuning.

Contents

Introduction 2
Components of the measurement scenario 4
The SPECweb2005 Benchmark and its process 5
Four results of the PRIMERGY TX150 S6 6
System statistics 8
Literature 10
Contact 10

Introduction

The SPECweb2005 benchmark is the industry standard for the performance measurement of web servers. It is part of the repertoire of standard benchmarks - SPECcpu2006, SPECjbb2005, SAP SD 2-tier, TPC-E and TPC-H are further examples - with the help of which the performance of the servers of all reputable manufacturers are systematically characterized. A whole range of publications with SPECweb2005 are available for the PRIMERGY servers from Fujitsu. These results were frequently world records at the time of publication and impressively documented the particular suitability of PRIMERGY for this application.

SPECweb2005 stands out for its superior mapping of the features of real web servers. With great time & effort and the active cooperation of Fujitsu, the Standard Performance Evaluation Corporation (SPEC) has analyzed productive systems from the applications online banking, electronic commerce and support & service. The features encountered there - from probability distribution during access to more or less frequently selected pages, the weighting of the GET and POST access methods, questions concerning SSL encryption right through to the distribution of the transferred data volumes - were incorporated in three notably different load profiles, which were to be tested separately *Banking*, E*commerce* and *Support*. In this way, a complex benchmark was created but which is also particularly informative thanks to its complexity.

Since the announcement of SPECweb2005 in June 2005, this potential has until now only been utilized to a limited extent in practice. If you look at the list of results, the one-sidedness of web server software and the method of generating dynamic pages are of particular note - throughout the publications of all manufacturers. The benchmark supports the interfaces PHP (PHP: Hypertext Preprocessor) and JSP (JavaServer Pages) for dynamic pages, whose contents are only prepared with current data at the time they are selected. Up to now publications have always used JSP. And the web server which has almost exclusively been used is *Rock Web Server* from Accoria Networks Inc. The reason behind the decision in favor of this combination is quite simple: it provides by far the highest values.

Comparability with real web servers falls by the wayside. Although the features of their HTTP traffic meticulously map the benchmark framework, the most frequently used software products are not used for the measurements. The following table shows the top positions in the Netcraft survey of November 2008 on the market shares of web servers[1]. The four products mentioned already represent a total market share of 93%. However, Rock Web Server also does not appear in the complete list of the approximate 50 products with at least 5,000 supported host names.

Manufacturers	Web servers	Number of sites November 2008	Market share
Apache	Apache	34,368,916	47.87%
Microsoft	IIS	23,740,478	33.07%
Google	GFE	812,220	11.31%
lighttpd	lighttpd	152,670	0.21%

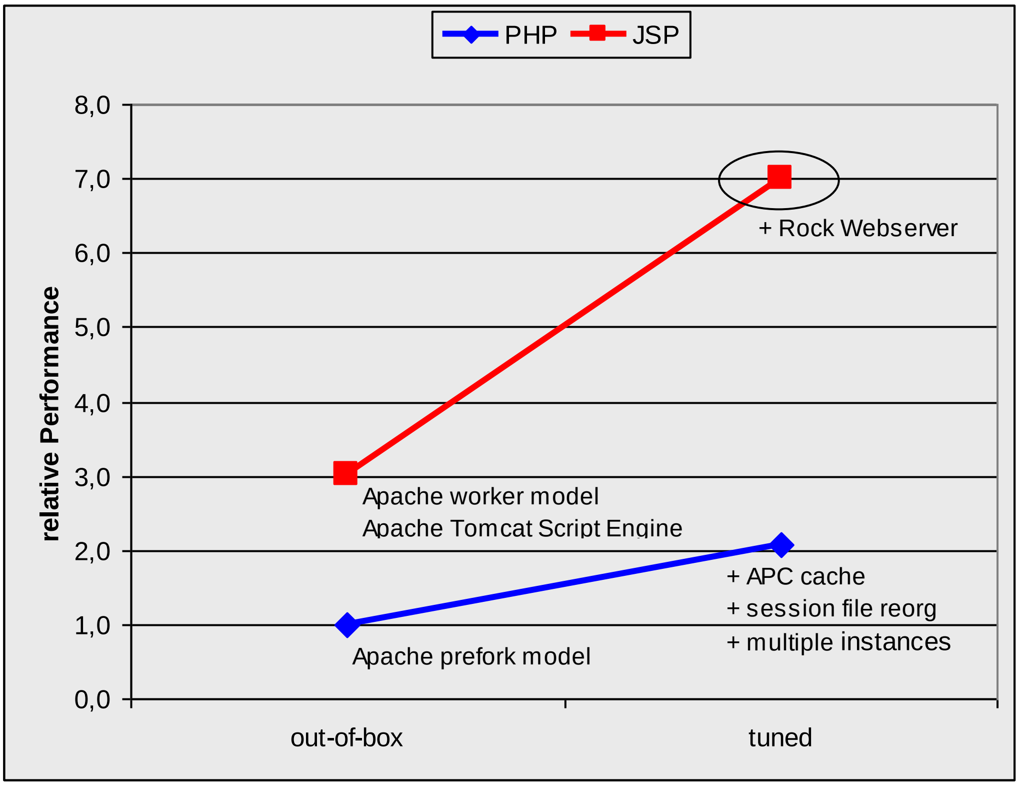
Four SPECweb2005 results for the PRIMERGY TX150 S6 are presented in this white paper. This of course also includes the obligatory top result with the Rock Web Server. This top value is the best-in-class result of all SPECweb2005 publications for x86-based monoprocessor systems, which are especially suitable as web servers[2]. The result particularly documents the performance of the I/O subsystem of the PRIMERGY TX150 S6 with a throughput of more than 1000 MB per second for disks and network. Incidentally, the PRIMERGY TX150 S6 also holds top positions when it comes to performance measurements for energy efficiency (SPECpower_ssj2008) and Java performance (SPECjbb2005).

According to the table the Apache HTTP server is the most widely spread web server product. Fujitsu has taken this market importance of Apache as a reason to add three further measurements, which all use Apache, to the top result of the PRIMERGY TX150 S6 with Rock Web Server. The Rock result is in a manner of speaking only the tip of the iceberg with regard to the scope for software configurations that are feasible for SPECweb2005 and of interest to users. We have replaced the high-performance, but exotic product with the most common product. We have also generated comparative values in two respects. Two of the four results use PHP for dynamic pages and two use JSP. Furthermore, for every tuned top value we have generated an out-of-the-box result, which only makes configuration adjustments that are absolutely

[1] see: http://news.netcraft.com/archives/2008/11/19/november_2008_web_server_survey.html
[2] see http://www.spec.org/web2005. All comparative statements in this document are valid as of January 16, 2009.

necessary and knowingly does without any tuning. These comparisons are to make it clear to the user what margin approximately exists when you invest in software tuning.

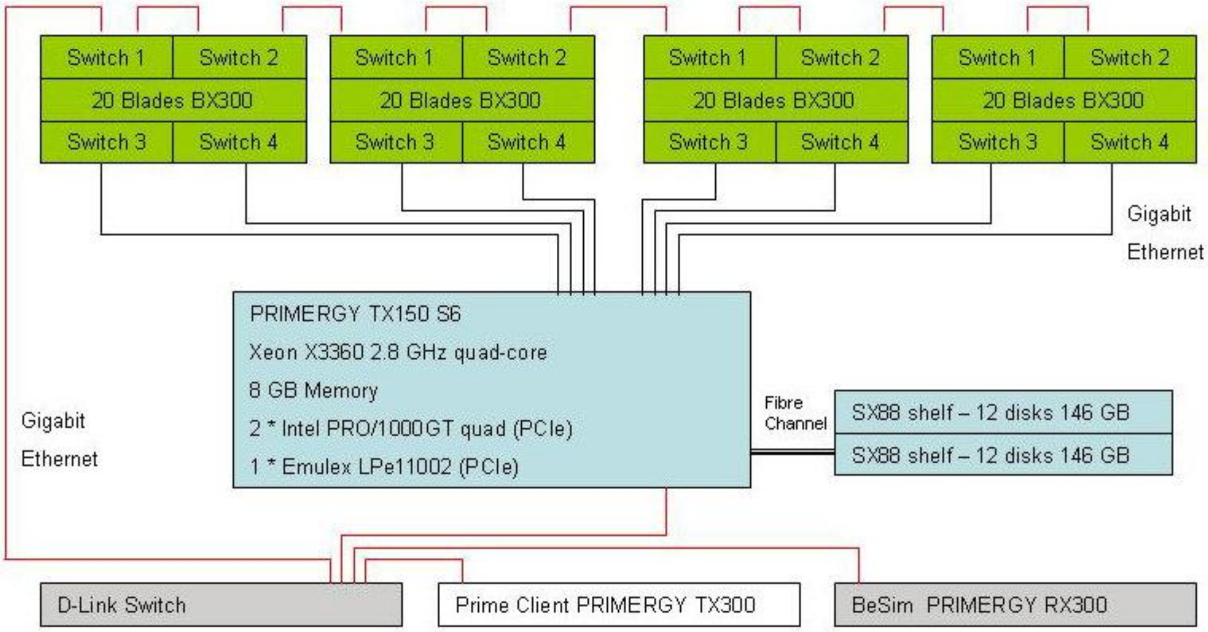
The following diagram shows this configuration scope and the prevailing scale, whereby the initial result with the lowest performance sets the standard. The two results with PHP are marked in blue and those with JSP are in red. The out-of-the-box result is in each case the starting value on the left and the line leads to the optimized result on the right, which is the best possible value. The features of the four measurement points indicated in the diagram are explained in detail in the following sections. In short the conclusion is as follows. Implementation in JSP has a three-fold higher performance than in PHP, in other words the simpler product has a higher resource requirement. In both cases, with PHP and JSP, tuning results in approximately a doubling of the comparable output performance. This is both very little and a great deal at the same time. The findings document the robustness of the Apache HTTP server, which also achieves an acceptable performance out-of-the-box. With software that is more sophisticated or challenging to operate (database server to name just one example) tuning successes may be more than twice as high. At the same time a two-fold increase provides a scope that definitely justifies the tuning outlay for the operator of a web server.



The following section presents the measuring environment of these four publications, or to be more exact the hardware used. Essentially, the environment was not changed, merely the number of client systems was varied according to the performance achieved. A brief outline of the SPECweb2005 benchmark now follows. Thus the prerequisite is created to present the four results in detail, including the measures that enabled the successive increases in performance. For those with a more technical interest the statements made there are ultimately underlined on the basis of statistics from the Linux operating system.

Components of the measurement scenario

The following diagram shows the components of the measurement scenario and how they are networked. Focus is placed on the *System under Test* (SUT) which is marked in blue. This is the web server PRIMERGY TX150 S6 including its storage subsystems, two FibreCAT SX88 rack modules with 24 disks, which are connected to the server via *Fibre Channel*. Apart from the components, the diagram also shows how they are physically networked. The connecting line between server and disks is the only line with the denotation Fibre Channel. All the other lines are denoted *Gigabit Ethernet*.


Switch 1
Switch 2
20 Blades BX300
Switch 3
Switch 4
Gigabit Ethernet
PRIMERGY TX150 S6
Xeon X3360 2.8 GHz quad-core
8 GB Memory
2 * Intel PRO/1000GT quad (PCIe)
1 * Emulex LPe11002 (PCIe)
Gigabit Ethernet
Fibre Channel
SX88 shelf – 12 disks 146 GB
SX88 shelf – 12 disks 146 GB
D-Link Switch
Prime Client PRIMERGY TX300
BeSim PRIMERGY RX300

The green level is formed by the clients, a total of 80 blades of type PRIMERGY BX300. Each Blade is connected to two of the four internal switches of the PRIMERGY BX300 housing, either switches 1 and 3 or switches 2 and 4. In this way, the connection between clients and web server comes into being first. There are eight subnets of type Gigabit Ethernet, which each connect ten clients to the server. The eight interfaces required in the server are provided by two Intel PRO/1000GT quad Ethernet cards in PCIe slots. The PRIMERGY TX150 S6 has three PCIe slots and the Emulex Fibre Channel HBA occupies the third slot.

An onboard Gigabit Ethernet interface integrates the PRIMERGY TX150 S6 into a further subnet, which exists in addition to the eight already mentioned (marked in red). All the components are represented in this subnet: apart from the web server and the 80 clients, a so-called back-end simulator (BeSim, a PRIMERGY RX300) and the prime client PRIMERGY TX300. The prime client controls the measurement process. It initializes the components involved, gives the clients the signal to start and end and collects and administers the result data. In contrast to the other components, it does not influence the performance measured. Thus the subnet marked in red is an administration LAN with its second function being the connection between web server and BeSim. Real web servers are not self-sufficient, they access downstream database or application servers in order to edit dynamic page contents or forward user input. In SPECweb2005 these operations are simulated by BeSim. The benchmark does not contain any real application parts, merely the communication between the web server and such a subsystem is reproduced.

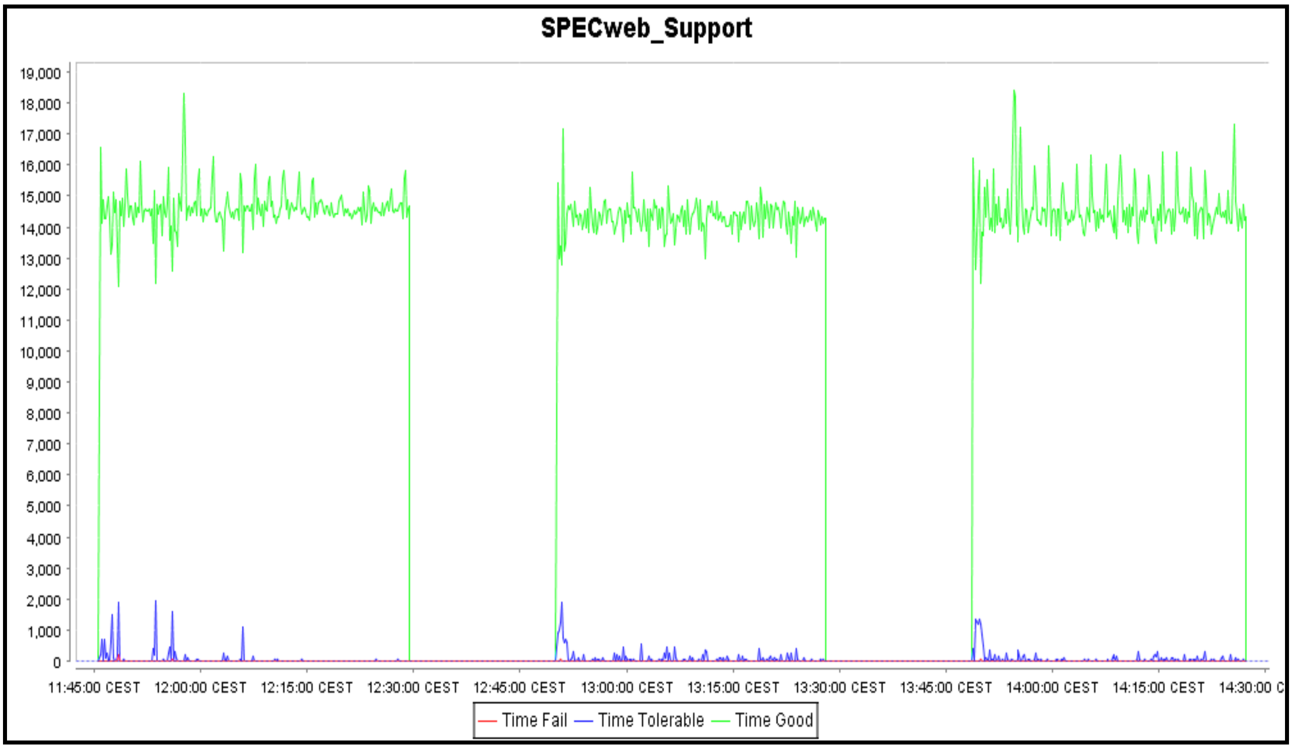
Apart from the hardware, all four SPECweb2005 results of the PRIMERGY TX150 S6 have the operating system *Red Hat Enterprise Linux* (RHEL) 5.1 in common for the SUT. Linux is very well suited for the web server application and is widespread.

The SPECweb2005 Benchmark and its process

This section provides some brief background information about SPECweb2005, which is required in order to understand the four benchmark results.

A complete measurement lasts about nine hours, three hours per load profile. The sequence of the load profiles is at the discretion of the tester. To ensuring reproducibility there is a cycle of three runs or iterations (each lasting just under one hour) for each load profile, which are made up of the sequence rampup, warmup, measurement interval and rampdown. The magic number per load profile is the number of simultaneously supported *sessions.* According to this number Java threads are started on each client during rampup, which continuously create requests and send them to the web server. This is continued during the warmup, in which the server is to reach a stable system load, and during the 30-minute measurement interval. During the rampdown the connections are released again and the Java threads are stopped. Waiting times, which are on average ten seconds for Banking and Ecommerce and five seconds for Support, are adhered to between the requests. An continuous sequence of requests and responses is denoted as a session. One session is processed per thread and after its conclusion a new session is started. This process is intended to reproduce the behavior of the users of a web site, who open pages, make HTTP requests and exit the site again.

For each requested page the response time until the page has fully built up, including embedded image files, is recorded and evaluated as follows. If the response time is below two seconds, the criterion GOOD is met; if it is less than four seconds, the criterion is TOLERABLE, otherwise the site is given a FAIL evaluation. With regard to the evaluation criteria there is an exception for the downloading of large files in the Support load profile. Here the criterion GOOD is met if a transfer rate of at least 99,000 bytes per second is reached. Accordingly a rate of 95,000 bytes per second applies for TOLERABLE. The benchmark run is valid if at least 95% of the requirements for each page is evaluated with GOOD, and 99% with at least TOLERABLE. These criteria, which decide the validity of a measurement, are called quality of service (QoS) criteria.



In order to illustrate the activity of the SUT the diagram shows the approximate 3-hour course of a complete run of the Support load profile with 13,604 configured sessions. The rhythm of the three measurement phases with intermediate breaks can clearly be seen. The number of requests completed with the criterion GOOD was counted every ten seconds for the green (top) line, and the blue (bottom) line accordingly shows the requests that were only evaluated with TOLERABLE. The third, red curve of FAILS is almost identical with the x axis, that is to say hardly any FAILS occurred. The total of all the requests in a counting period of ten seconds is somewhat larger than the number of sessions, because the average runtime of a request, including waiting and response time, is less than ten seconds and on average a little more than one request was handled per session. The runs for the load profiles Banking and Ecommerce look similar to the diagram above. Thus a complete measurement of the SPECweb2005 benchmark lasts about nine hours.

To date we have spoken of pages in general. To a lesser extent these are static pages and to a larger extent dynamic ones, for which implementations are alternatively available in PHP and JSP. The pages contain embedded image files, which in keeping with common practice are transferred via a second, parallel HTTP connection per session. Browser caching effects are simulated through the use of *If-Modified-Since* requests. The Banking load profile uses SSL-encrypted HTTPS traffic all the time, and the Support load profile only uses HTTP without encryption. The first steps per session for Ecommerce are effected via HTTP, then - when order and payment operations are emulated - the transfer is SSL-encrypted as with Banking.

Four results of the PRIMERGY TX150 S6

The four SPECweb2005 results according to the following table were published for the PRIMERGY TX150 S6 as a SUT in the second half-year of 2008.

	#1	**#2**	**#3**	**#4**
Result	**PHP**	**PHP**	**JSP**	**JSP**
	out-of-box	**tuned**	**out-of-box**	**tuned**
SPECweb2005	**2641**	**5457**	**7945**	**18495**
SPECweb2005_Banking	2700	6240	9600	29040
SPECweb2005_Ecommerce	3600	7360	10640	30480
SPECweb2005_Support	4200	7840	10880	15840
Increase factors:				
SPECweb2005_Banking	N/A	2.3	1.5	3.0
SPECweb2005_Ecommerce	N/A	2.0	1.4	2.9
SPECweb2005_Support	N/A	1.9	1.4	1.5
#Subnets	2	4	8	8
#Clients	20	40	80	80
Client Operating system	Windows XP	Windows XP	Windows XP	Windows XP
HTTP Server	Apache 2.2.9	Apache 2.2.9	Apache 2.2.3	Rock 1.4.6
#HTTP Server Instances	1	2	1	1
Apache Process Model	prefork	prefork	worker	N/A
Script Engine	PHP 5.1.6	PHP 5.2.6	Tomcat 5.5.20	Rock JSP 1.3.1
Server Cache	N/A	APC 3.0.19	N/A	N/A
Script Method	PHP	PHP	JSP	JSP
SUT Operating system	RHEL 5.1	RHEL 5.1	RHEL 5.1	RHEL 5.1
Published	Aug-2008	Nov-2008	Aug-2008	Jun-2008

The table shows the results in ascending order according to the primary metric SPECweb2005. As listed in the last section, the maximum number of sessions, for which the QoS criteria are still met, is determined for each load profile. These are the three secondary metrics SPECweb2005_Banking, SPECweb2005_Ecommerce and SPECweb2005_Support. These values are standardized to a reference system for the primary metric SPECweb2005 and the geometric mean is formed. Of the various possibilities of forming a mean value the geometric mean stands out because the influence of individual values on the mean value does not depend on their absolute size: the domination of the final result by an excellent individual result is prevented. The table provides the details of the four datapoints that were already shown in the diagram on page 3.

The table shows the improvement compared with the previous measurement on the left as the increase factors for each load profile. The increase for Support is consistently the lowest and for Banking the largest.

The inconsistency in the improvements proves the difference in the load profiles. Where the differences are from a technical viewpoint is explained in the following section on the basis of system statistics.

Measurement #1 uses Apache and PHP out-of-the-box. Adaptations have been made in the *httpd.conf* configuration file of Apache, which are needed for the operability of the benchmark with the specified number of sessions and for the activation of PHP. Adaptations have not been made at all in the *php.ini* configuration file. PHP is not thread-compatible; the *libphp.so*, which is to be loaded dynamically in Apache, is not threadsafe. For this reason only the prefork process model from Apache, where a separate process exists for each active connection, is eligible for all the measurements that use PHP.

The tuning, which results in measurement #2, is a package consisting of the following four measures:

- PHP is a script language. The implementation of a PHP page requires parsing and the generation of an executable interim code. This compilation is effected as standard each time a page is requested. There are PHP extensions, which enable caching and reuse of compiled pages. The *Alternative PHP Cache* (APC) is such an extension. The improvement in all three load profiles amounted to about 50%.
- PHP, as all applications for dynamic pages, implements sessions[3] as a method of putting successive HTTP requests together to form a unit, which enables reference to be made to previous user input. PHP implements this with one file per session. As standard these files are in a single subdirectory. This can be equalized, and reorganization can also be simplified. This measure benefits Banking and Ecommerce, whereas Support does not use any sessions.
- One focal point of the configuration of a web server is the alignment with the available physical memory. The memory should be fully used, but an appropriate share must be taken into account for the cache of the file systems. At the same time paging and swapping must by all means be avoided. The Apache parameters for the maximum number of simultaneous connections permit a fine adjustment of these tradeoffs for each load profile.
- In measurement #2 the load was distributed over two instances of Apache. During this measurement the number of processes was already so high that two instances have notably relieved the load of the semaphores, with whose help the processes of an Apache instance are synchronized.

Measurement #3 shows the out-of-the-box result with Apache and JSP. For this purpose, the product *Apache Tomcat* is needed in addition to the Apache HTTP server. The JSP-compatible Tomcat is not integrated in the HTTP server as a dynamic library, as with PHP, but exists as a separate instance in the form of a Java Container, to which the HTTP server forwards requests for JSP pages. The HTTP server itself only processes static pages in this construct. For measurement #3 only the most necessary adjustments were again made in the configuration files in order to run the benchmark with the user numbers stated in the table. However, other than for the measurements with PHP the higher-performance worker process model from Apache can be used here, in which only a single thread exists for each active connection.

The scope for tuning the Apache / Apache Tomcat construct is small in comparison with Apache / PHP. Changing to Java entails an enormous reduction in the memory requirements per active connection. Consequently, it becomes possible to keep two Apache threads (according to the two connections for HTTP pages and embedded image files) permanently in the main memory for each configured SPECweb2005 session and to use *Persistent Connections* all the time. Persistent connections are not released after completion of a request, but continue to exist until the next request from the same user or a configurable time limit expires. On the other hand, Apache / PHP calls for memory-related multiplexing for the mapping of the users to the available work processes; an abundant field of tuning that is unnecessary for Apache / JSP from the outset.

The enormous increase that is realized with the last measurement #4 requires the other server architecture that the Rock Web Server offers compared with Apache / Apache Tomcat: the Java Container runs in the web server, which makes communication between the processes no longer applicable. Furthermore, the Rock Web Server does not need a separate thread or process per active connection, but manages with a comparatively low number of work processes. This results in additional memory relief in favor of the file systems cache and in a lower process administration overhead.

The fact that the version differences stated in the table did not have any influence on the performance measured for the Open Source products Apache and PHP should still be documented. Here it is only a

[3] This now refers to a language feature of PHP, which must strictly speaking be distinguished from the use of the term in conjunction with the user model of the SPECweb2005 benchmark.

matter of whether the version delivered with RHEL 5.1 was used or, in case of recompilation, the latest version available on the Internet. For example, the integration of APC calls for recompilation.

If a concise conclusion is to be drawn from the four measurements presented here, the two rules of thumb already mentioned in the introduction are suitable. With identical tuning quality the difference between the PHP and JSP results is roughly three-fold. In other words, JSP is approximately three times higher in performance than PHP in terms of lower resource consumption. And the scope for tuning in both cases, PHP and JSP, was two-fold in scale. These findings gained from benchmark results can be transferred to real applications as follows. The differences in resource consumption between PHP and JSP should be similar in other applications than SPECweb2005. With scope for tuning on the other hand, the benchmark shows if anything an upper limit, because according to code extent and programming language resources used the dynamic pages of the benchmark are less complex. Many of the tuning gains have a once-only impact on the page requested and are of greater consequence with simply structured pages.

System statistics

The statements made in the last section about the four results should now be mirrored with the statistics of the Linux operating system. All the previous verbal descriptions are to be found again in these figures, concentrated in one column per measurement and load profile in the table on the following page. Here we are dealing with the technical viewpoint of web server administrators with regard to the systems they support.

The top half of the table contains absolute values of throughputs as well as memory and CPU load. And the important number of processes and threads that exist on average during the measurement interval, which shows the difference in the architecture between Apache and Rock in a particularly impressive way. The throughput rates for web pages are determined from the view of the clients. Since each page contains embedded image files, the rate of HTTP requests determined on the server is considerably higher. In the last block of the top half of the table you can see the I/O throughputs, each with totals for read and write. Essentially, these throughputs are proportional to the achieved measurement result and achieve in the Support load profile of measurement #4 an altogether impressive total size for a monoprocessor of more than 1,000 MB per second. SPECweb2005 is thus a genuine system benchmark that fully incorporates the I/O subsystem.

The bottom half of the table contains details per requested web page, first the CPU consumption per page in milliseconds, then the number of process changes, interrupts and page faults. Followed by the I/Os per page. As expected, specifically the values for the network I/Os for all four measurements are about the same. What needs to be transferred per web page is defined by the benchmark and cannot be influenced by the selected software stack or the tuning level.

The step from measurement #1 to #2 can be best understood if you look at the CPU consumption and the page faults. You can also see how the APC cache reduces the memory requirements per work process. This reduction is greatly continued in the transition to JSP in measurement #3. The relief in the load on memory consumption enables the clear increase in the number of processes and threads. In measurement #3 the situation is reached where for each session two HTTP connections and thus two Apache threads can be permanently stored in the main memory.

On account of the architecture this picture changes completely in measurement #4. Rock does not need any user-specific work processes and in comparison with measurement #3 manages with a number of threads that is two orders of magnitude as small; the reductions in the process changes, interrupts and page faults per web page are accordingly dramatic.

Result	**#1 PHP out-of-box**			**#2 PHP tuned**			**#3 JSP out-of-box**			**#4 JSP tuned**		
Profile	**bank**	**ecom**	**supp**	**bank**	**ecom**	**supp**	**bank**	**ecom**	**supp**	**bank**	**ecom**	**supp**
#sessions	2700	3600	4200	6240	7360	7840	9600	10640	10880	29040	30480	15840
pages per sec	409	337	414	903	680	766	1458	957	1079	4083	2805	1555
http requests per sec	4908	5727	8725	10836	11553	16141	17492	16251	22727	48999	47656	32769
#processes or threads	1728	2488	2671	3680	4729	4479	23271	30590	25704	240	356	245
%memory used	99	85	99	99	99	99	99	99	99	99	99	99
file system cache (MB)	4089	2275	3604	1952	590	1496	648	1124	3943	2970	2345	1763
process data (MB)	3762	4577	4526	5950	7520	6640	7421	6961	4188	4580	5635	6366
MB data per process	2.18	1.84	1.69	1.62	1.59	1.48	0.32	0.23	0.16	19.08	15.83	25.98
%user mode	66	59	52	61	49	45	66	64	52	64	46	13
%system mode	27	25	29	33	44	52	32	32	44	36	51	76
%wait for I/O	1	1	3	2	2	2	1	1	2	0	2	5
%idle	5	15	16	4	6	2	1	3	2	0	0	6
MB/s total disk I/O	1.08	0.77	56.45	3.48	2.44	155.32	3.75	2.39	161.49	9.53	6.79	194.96
MB/s total network I/O	16.53	52.29	238.51	36.89	106.87	437.38	54.47	147.73	617.74	150.63	440.74	888.72
per requested page:												
millisecs user mode	6.44	7.03	5.03	2.69	2.86	2.33	1.82	2.67	1.92	0.63	0.66	0.33
millisecs system mode	2.69	2.93	2.83	1.48	2.60	2.69	0.87	1.34	1.64	0.35	0.73	1.95
millisecs total	9.13	9.95	7.86	4.17	5.46	5.02	2.69	4.01	3.56	0.98	1.39	2.29
context switches	26.77	34.04	36.32	21.88	27.89	30.14	22.36	30.51	30.29	2.49	7.33	9.14
interrupts	22.45	24.03	22.35	20.20	22.11	22.95	17.76	24.73	26.11	6.91	10.64	19.37
page faults	65.40	131.89	69.19	8.66	37.08	10.55	8.59	6.82	3.70	0.03	0.02	0.41
disk reads	0.19	0.13	0.73	0.29	0.22	1.25	0.37	0.23	0.97	0.32	0.22	0.77
network packets rx	43.62	72.45	142.22	43.99	72.85	140.20	25.49	56.19	124.26	25.43	55.49	128.64
network packets tx	50.92	126.27	416.18	53.71	129.18	413.10	34.76	113.14	399.64	42.97	117.99	401.95
avg KB network read	5.17	8.36	12.67	5.19	8.74	12.48	3.59	7.64	11.26	3.57	8.80	11.66
avg KB network write	35.23	146.75	563.38	35.66	148.42	558.51	33.78	146.80	561.51	33.33	148.33	559.83

Literature

PRIMERGY Systems	http://www.ts.fujitsu.com/primergy
PRIMERGY TX150 S6	http://www.ts.fujitsu.com/products/standard_servers/tower/primergy_tx150s6.html
PRIMERGY Performance	http://www.ts.fujitsu.com/products/standard_servers/primergy_bov.html
SPECweb2005	http://www.spec.org/web2005
	Benchmark Overview SPECweb2005 http://docs.ts.fujitsu.com/dl.aspx?id=efbe8db4-7b1b-481e-bdee-66bdfa624b57

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://www.ts.fujitsu.com/primergy

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