

УДК 004.738.5:004.051

Е.С. Васильченко, А.В. Горбенко

Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков

АНАЛИЗ ПРОИЗВОДИТЕЛЬНОСТИ СИСТЕМ УПРАВЛЕНИЯ САЙТОМ CMS

Проанализированы существующие системы управления сайтом, а также средства для нагрузочного и стрессового тестирования производительности. Выполнено исследование производительности систем CMS, написанных на языке PHP: Drupal, PHP-Nuke и Joomla.

Ключевые слова: CMS, производительность, PHP, JMeter.

Введение

На сегодняшний день Интернет играет важную роль в жизни людей и развитии бизнеса. Наличие Web-сайта является неотъемлемой составляющей успеха любой компании. Одним из наиболее эффективных способов создания крупных Web-сайтов (Web-порталов) на сегодняшний день является использование CMS. Аббревиатура CMS расшифровывается как Content Management System, или система управления содержимым (контентом) сайта. Также CMS иногда называют «двигателем» (или «движком») сайта (site engine) [1]. CMS предоставляет готовый шаблон web-сайта и реализует функции автоматизации процесса модернизации сайта, управления его содержимым, добавления, удаления и редактирования Web-страниц. Некоторые CMS также позволяют легко расширять функциональность интернет-ресурса, менять дизайн, добавлять новые интерактивные модули (ленты новостей, фотогалереи, рекламные баннеры, каталоги и т.д.). Существует большое множество CMS, написанных на разных языках, имеющих разное предназначение и характеристики. Правильный выбор CMS – залог успеха разработки Web-сайта, однако, такой выбор является нетривиальной задачей. Одними из важнейших критериев выбора являются производительность, характеризующаяся скоростью (задержкой) обработки запроса пользователя и масштабируемостью – возможность одновременного обслуживания большего количества запросов без существенного снижения производительности CMS. Для тестирования и оценки характеристик производительности Web-сайтов существует набор специальных утилит, моделирующих нагрузку на Web-сайт и оценивающих показатели производительности.

Цель данной работы – обзор «рынка» CMS, анализ средств оценивания их производительности, а также проведение сравнительного анализа производительности нескольких популярных CMS.

Анализ CMS-систем для создания Web-порталов

Для разработки CMS-систем используются различные технологии и языки программирования,

такие как PHP, Java, ASP.NET, Perl, Python, Ruby и др.

В результате проведенного анализа, а также с учетом данных, представленных в [2] было установлено, что наиболее популярным языком создания CMS систем являются PHP и Java.

Относительное соотношение языков программирования, используемых для создания 25 наиболее популярных CMS-систем показано на рис. 1.

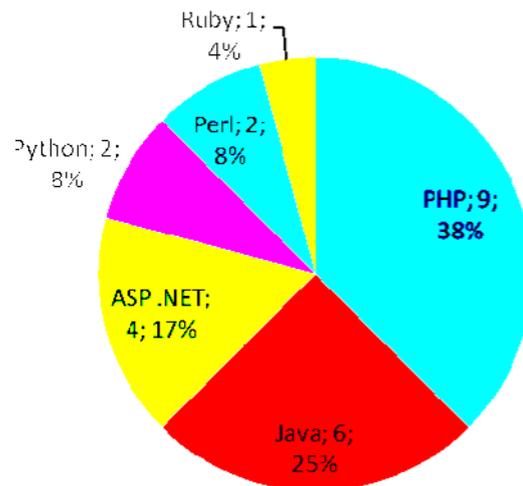


Рис. 1. Распространенность языков программирования для CMS систем

В табл. 1 представлен список наиболее популярных CMS-систем, распространяемых бесплатно, а также указан используемый для их создания язык программирования.

Для работы CMS-системы, как и для любого Web-сайта, необходимо наличие Web-сервера и/или сервера приложений, в зависимости от технологии, использованной для разработки CMS-системы.

Также для организации хранения различных данных необходимо наличие СУБД.

Например, для CMS-систем, созданных с использованием PHP и Java традиционным является использование следующих системных компонент: Web-сервер Apache; сервер приложений Apache Tomcat; СУБД MySQL.

Популярные CMS-системы

Название CMS	Версия	Язык программирования	Лицензия	Web-сайт
Php-Nuke	8.1	PHP	GNU GPL	www.phpnuke.org
Mambo	4.6	PHP	GNU GPL	www.mambo-foundation.org
drupal	6.14	PHP	GNU GPL	www
php fusion	7	PHP	Affero GPL	www.php-fusion.co.uk
openems	7.5	Java	GNU LGPL	www.opencms.org
Hippo CMS	7.1	Java	Apache	www.onehippo.com
liferay	5.2	Java	MIT	www.liferay
cuyahoga	1.6	ASP.NET	BSD License	www.cuyahoga-project.org
DotNetNuke	5.1	ASP.NET	BSD	www.dotnetnuke.com
Sense/Net	6.0	ASP.NET	GNU GPL	www.Sensenet.hu
PyLucid CMS	0.8.6	Python	GNU GPL	www.PyLucid.org
RefineryCMS	0.9	Ruby	MIT	www.refinerycms.com
Bricolage	1.10	Perl	BSD	www.bricolagecms.org
radiant (cms)	0.8.1	ruby	MIT	www.radiantcms.org
WebGUI	7.8	Perl	GNU GPL	www.webgui.org
jAPS	2.0	Java	GNU GPL	www.japsportal.org
Alfresco	3.2	Java	GNU GPL	www.alfresco.com
Nuxeo	5.3.0	Java	GNU LGPL	www.nuxeo.org
Joomla!	1.6	PHP	GNU GPL	www.joomla.org
Concrete5	5.3	PHP	MIT	www.concrete5.org
Cms Made Simple	1.6	PHP	GNU GPL	www.cmsmadesimple.org
eZ Publish	4.2	PHP	GNU GPL	www.ez.no
SilverStrike	2.3.4	PHP	BSD License	www.silverstrike.org
Plone	3.3	Python	GNU GPL	www.plone.org
MojoPortal	2.3	ASP.NET	CPL	www.mojoportal.com

Анализ средств для проведения нагрузочного и стрессового тестирования

Для Web-приложений тестирование производительности является таким же важным этапом разработки, как и тестирование функциональных характеристик.

Основанные на Web-технологиях средства тестирования производительности позволяют компаниям проверять быстродействие и стабильность работы своих Web-сайтов и предпринимать меры по их усовершенствованию до начала эксплуатации.

Наиболее часто используемыми на практике средствами тестирования производительности Web-приложений являются:

- 1) Jakarta JMeter;
- 2) WebLoad;
- 3) SilkPerformer.

Указанные продукты различаются как по цене, так и по гибкости создания тестовых сценариев, разнообразию типов проверяемого контента, обратной связи в ходе проведения нагрузочных тестов, глубине анализа и содержательности отчетов.

Общим свойством таких систем является возможность автоматизированного создания сценариев тестирования при помощи браузера, наличие средств динамического регулирования тестовой нагрузки и создания отчетов о производительности, а также эмуляция запросов от множества виртуаль-

ных клиентов с множества компьютеров при помощи удаленных агентов.

Чаще всего, утилиты тестирования производительности интегрируются с серверными средствами контроля, например Performance Monitor для ОС Windows, которые дают полезные сведения об использовании сервером системных ресурсов, и могут быть использованы для выявления программных ошибок, приводящих к «утечкам» памяти, «зависанию» или блокировке процессов и др.

При работе с утилитой WebLoad 4.0, разработанной фирмой RadView, наиболее интересными особенностями являются тестирование, выявляющее достижение порога какой-либо конкретной характеристики, и выдача результатов непосредственно в ходе выполнения теста, а также использование JavaScript для создания гибких динамических тестовых сценариев.

SilkPerformer 4.0 компании Segue Software выделяется удобным графическим интерфейсом и особенно подходит для тех случаев, когда проверяющие не являются профессиональными тестировщиками. SilkPerformer удобен также тем, что дает возможность пробного запуска теста для выявления ошибок в сценарии.

В данной работе в качестве средства для проведения тестирования было решено выбрать Jakarta JMeter [3]. Jakarta JMeter представляет собой инструмент для проведения нагрузочного тестирования, разрабатываемый Apache Jakarta Project. Хотя изна-

начально JMeter разрабатывался как средство тестирования web-приложений, в настоящее время он способен проводить нагрузочные тесты для JDBC-соединений, FTP, LDAP, SOAP, JMS, POP3, HTTP и TCP. Интересна возможность создания большого количества запросов с помощью нескольких компьютеров при управлении этим процессом с одного из них. В программе реализованы механизмы авторизации виртуальных пользователей, поддерживаются пользовательские сеансы. Организовано логирование результатов теста и разнообразная визуализация результатов в виде диаграмм, таблиц и т.п. Существенным достоинством JMeter является то, что это свободно распространяемое ПО с открытым исходным кодом.

Тестирование производительности CMS-систем на основе PHP

В данной статье представлены результаты тестирования производительности трех популярных CMS-систем, написанных на языке PHP: Drupal, PHP-Nuke и Joomla. Тестирование проводилось при помощи Jakarta JMeter. При тестировании фиксировались следующие параметры:

- среднее время ответа сервера, t_{cp} ;
- среднее количество запросов, обрабатываемых в секунду, r_{cp} ;
- количество успешно обработанных запросов, R_y ;
- количество запросов, завершившихся ошибкой, R_o .

Основной целью экспериментов являлось нахождение количества одновременных обслуживаемых пользователей, при котором наблюдается максимальная производительность CMS-системы, а также определение количества одновременных пользователей, приводящих к перегрузке сервера (т.е. к экспоненциальному росту времени обработки и, в конечном итоге, прекращению обработки запросов).

Для этого с каждой CMS-системой было проведено тридцать четыре эксперимента с разным количеством одновременных пользователей, обращающихся к Web-серверу, начиная от одного и заканчивая сотней пользователей с шагом увеличения, равным трем. Для определения средних характеристик производительности каждый пользователь в каждом эксперименте выполнял десять последовательных запросов.

Так как целью эксперимента было определение наиболее производительной CMS-системы, то использовался сравнительно простой сценарий тестирования – загрузка начальной страницы Web-сервера общим объемом 200 Кбайт, содержащей текст и картинку в формате jpeg.

Очевидно, что тестирование всех CMS-систем

для последующего сравнения их производительности и выбора лучшей должно выполняться на компьютерах с одинаковой аппаратной и программной конфигурацией. В данном случае, сервер, использованный для проведения тестирования производительности, имел следующую конфигурацию:

- конфигурация аппаратного обеспечения: процессор Intel Core 2 Duo E8200 2,66 GHz, 4 GB ОЗУ;
- конфигурация системного программного обеспечения: ОС Windows 7 Ultimate x64, Web-сервер Apache 2.2.17, PHP 5.3.3, СУБД MySQL 5.1.51.

Клиентское ПО Jakarta JMeter было запущено на ноутбуке, непосредственно подключенном по сети Fast Ethernet к серверу (рис. 2).

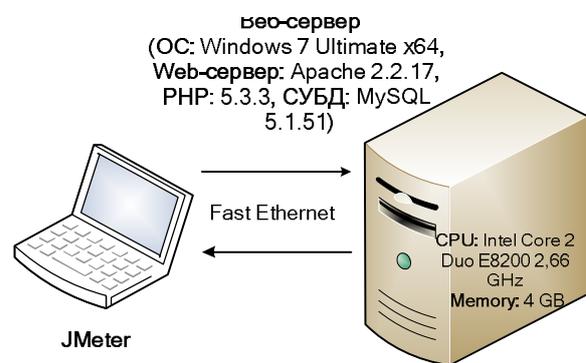


Рис. 2. Конфигурация системы тестирования CMS систем

Результаты тестирования CMS систем приведены в табл. 2 – 4.

На рис. 3 представлен график зависимости среднего времени ответа сервера от количества одновременных пользователей.

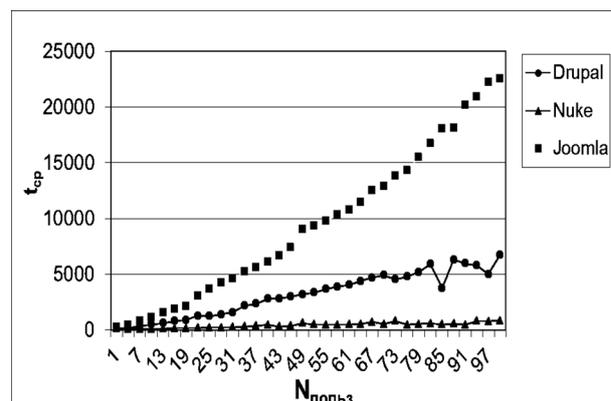


Рис. 3. График зависимости среднего времени ответа сервера от количества одновременных пользователей

График зависимости среднего количества запросов, обрабатываемых в секунду, от количества пользователей приведен на рис. 4.

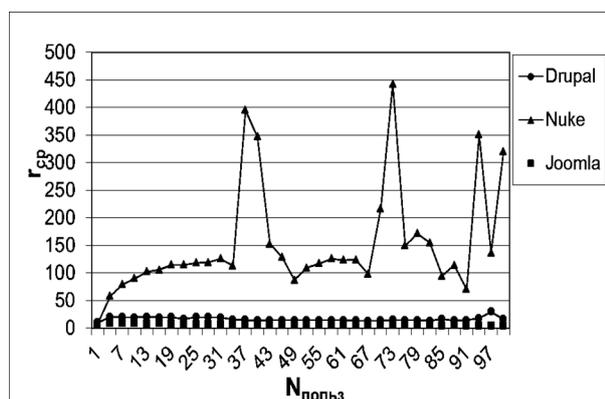


Рис. 4. График зависимости среднего количества запросов, обрабатываемых в секунду, от количества одновременных пользователей

Таблица 2

Результаты тестирования CMS Drupal

$N_{польз}$	$t_{ср}$, мс	$r_{ср}$, запр./с	R_y	R_o
1	113	9,56938	10	0
4	199	20,25316	40	0
7	337	20,03434	70	0
10	505	19,04399	100	0
13	616	20,07412	130	0
16	801	19,11132	160	0
19	892	19,69933	190	0
22	1269	16,60377	220	0
25	1238	19,85388	250	0
28	1387	19,72803	280	0
31	1595	19,20813	310	0
34	2190	15,33051	340	0
37	2356	14,84275	370	0
40	2803	13,34312	400	0
43	2829	13,89383	430	0
46	2975	14,19446	460	0
49	3197	14,43682	490	0
52	3370	14,45127	520	0
55	3660	14,0256	550	0
58	3886	13,87859	580	0
61	4060	14,23571	610	0
64	4402	13,72801	640	0
67	4705	13,18924	670	0
70	4931	13,28223	700	0
73	4541	14,85249	730	0
76	4786	14,39121	760	0
79	5195	14,01952	790	0
82	5924	13,16592	820	0
85	3772	16,83817	72	49
88	6331	13,2584	880	0
91	5972	14,01099	910	0
94	5806	17,90475	497	1
97	5022	29,81818	164	1
100	6747	16,08713	706	1

Таким образом, из полученных результатов тестирования можно сделать следующие выводы.

1. CMS PHP-Nuke показала наибольшую производительность и наилучшее время отклика. Минимальное время отклика составляло 67 мс. Максимальная пропускная способность CMS PHP-Nuke

достигается при обработке 25-30 одновременных запросов и лежит в диапазоне от 120 до 130 запросов в секунду. Однако дальнейшее увеличение количества одновременных запросов приводит к нестабильной работе Web-сервера и сбоям, которые проявляются в виде сообщений об ошибках при обработке значительной части запросов.

2. CMS Drupal показала меньшую производительность и большее время отклика по сравнению с PHP-Nuke, однако сбои в работе сервера проявлялись только при 85 и больше пользователях, одновременно обращающихся к Web-серверу. Минимальное время отклика от Web-сервера при использовании CMS Drupal составило 113 мс, а максимальная пропускная способность – 20,25 запроса в секунду, была достигнута уже при четырех одновременных запросах, однако при дальнейшем увеличении количества одновременных пользователей пропускная способность Web-сервера уменьшалась незначительно, а время обработки одного запроса возрастало фактически линейно.

Таблица 3

Результаты тестирования CMS Nuke

$N_{польз}$	$t_{ср}$	$r_{ср}$	R_y	R_o
1	111	8,7184	10	0
4	67	57,97101	40	0
7	78	78,82883	70	0
10	104	90,09009	100	0
13	119	102,3622	130	0
16	138	105,3325	160	0
19	158	114,873	190	0
22	185	114,5237	220	0
25	201	118,7085	250	0
28	219	119,1997	280	0
31	233	125,8117	310	0
34	298	112,9977	193	10
37	381	395,3488	17	20
40	479	347,2222	25	15
43	297	152,5424	18	25
46	384	129,0323	12	26
49	623	86,89459	61	2
52	487	109,3376	137	2
55	471	117,5657	170	1
58	464	125,8567	202	2
61	500	123,6383	227	1
64	524	124,3348	257	2
67	706	97,47292	135	2
70	541	216,4179	29	41
73	813	442,623	54	19
76	465	149,6599	22	54
79	541	171,6418	23	37
82	597	154,9296	22	55
85	511	94,2029	13	61
88	573	114,5038	15	73
91	463	70,42254	5	25
94	813	350,6494	54	40
97	766	136,4865	202	2
100	830	320,6751	76	24

Таблиця 4

Результаты тестирования CMS Joomla

N _{польз}	t _{ср}	Г _{ср}	R _v	R _o
1	236	4,66418	10	0
4	463	8,68621	40	0
7	824	8,34128	70	0
10	1163	8,41538	100	0
13	1548	7,7961	130	0
16	1884	8,58277	160	0
19	2161	8,11099	190	0
22	3044	7,07282	220	0
25	3716	6,2762	250	0
28	4234	6,41466	280	0
31	4610	6,29902	310	0
34	5234	6,23419	340	0
37	5651	6,1603	370	0
40	6124	6,20386	400	0
43	6649	6,01037	430	0
46	7437	5,90341	460	0
49	9036	5,15692	490	0
52	9362	5,08632	520	0
55	9792	5,16388	550	0
58	10338	5,07712	580	0
61	10763	5,21988	610	0
64	11492	5,12463	640	0
67	12563	4,97317	670	0
70	12940	5,00054	700	0
73	13846	4,88396	730	0
76	14340	4,97158	760	0
79	15545	4,87353	790	0
82	16784	4,67906	820	0
85	18098	4,56633	850	0
88	18143	4,64419	880	0
91	20190	4,30081	910	0
94	20919	4,37543	940	0
97	22245	4,26119	970	0
100	22546	4,34747	1000	0

3. CMS Joomla показала наибольшую стабильность, не приводя к сбоям в работе сервера даже при максимальной нагрузке в сто одновременных пользователей. Однако производительность и время отклика по сравнению с другими CMS оказались худшими.

Заключение

Существует множество систем управления сайтами CMS, написанных на разных языках. Задача выбора CMS является одной из важнейших, решение которой во многом будет определять эффективность и стабильность работы Web-сайта независимо от его наполнения.

Правильный выбор CMS по критерию производительности может базироваться на априорном тестировании и сравнении производительности нескольких альтернативных CMS с использованием специальных утилит, наибольшей популярностью среди которых пользуется JMeter.

В ходе тестирования трех CMS систем, написанных с использованием PHP, наибольшую производительность показала PHP-Nuke, а наилучшую стабильность – Joomla. CMS Drupal показала средние результаты по обоим показателям и может использоваться в качестве компромиссного решения задачи выбора эффективной CMS.

Список литературы

1. Что такое CMS? [Электронный ресурс]. – Режим доступа к ресурсу: http://www.internet-technologies.ru/articles/article_1303.html.
2. List of content management systems. [[Электронный ресурс]. – Режим доступа к ресурсу: http://en.wikipedia.org/wiki/List_of_content_management_systems.
3. Tutorial on testing webservices with Apache JMeter. [Электронный ресурс]. – Режим доступа к ресурсу: <http://www.testingminded.com/2009/01/tutorial-on-testing-webservices-with.html>.

Поступила в редколлегию 14.12.2010

Рецензент: д-р техн. наук, проф. В.С. Харченко, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков.

АНАЛІЗ ПРОДУКТИВНОСТІ СИСТЕМ УПРАВЛІННЯ САЙТОМ CMS

Є.С. Васильченко, А.В. Горбенко

Проаналізовані існуючі системи управління сайтом, а також засоби для навантаження і стресового тестування продуктивності. Виконано дослідження продуктивності систем CMS, написаних на мові PHP: Drupal, PHP-Nuke і Joomla.

Ключові слова: CMS, продуктивність, PHP, JMeter.

PERFORMANCE ANALYSIS OF CONTENT MANAGEMENT SYSTEMS

E.S. Vasil'chenko, A.V. Gorbenko

Existing content management systems are analyzed, and also facilities for loading and stressful testing performance. Research is executed performance systems CMS, written in language PHP: Drupal, PHP-Nuke and Joomla.

Keywords: CMS, performance, PHP, JMeter.