электронный научно-технический журнал
ИНЖЕНЕРНЫЙ ВЕСТНИК
Издатель ФГБОУ ВПО "МГТУ им. Н.Э. Баумана". Эл No. ФС77-51036. ISSN 2307-0595

# Имитационное моделирование Internet-систем, построенных с использованием системы управления контентом Joomla

# 03, март 2015

Афанасьев Г. И.[1,*], Ковалева Н. А.[1]

УДК: 004.738.5:004.91:004.94

[1]Россия, МГТУ им. Н.Э. Баумана
[*]gaipcs@bmstu.ru

Одним из существенных элементом современных Internet-систем, используемых для автоматизации процессов их развертывания, сопровождения, модификации и расширения являются системы управления контентом (CMS - content managmrnt system) [1, 2, 3]. На сегодняшний день используемых систем управления контентом насчитывается несколько десятков, из которых одной из наиболее популярной среди разработчиков является такая система как Joomla [1, 8, 13].

Если рассмотреть структуру этих Intrnet-систем, то можно выделить следующие основные составляющие. Это в первую очередь web-сервер. Наиболее популярными являются web-серверы Apache и Microsoft IIS [1,13] . Основной задачей web-сервера является прием поступающих запросов из Internet, и выдача обратно в Internet ответа в виде сформированных HTML страниц, содержащих HTML и Java-Script инструкции.

Следующая составляющая это серверный язык обработки клиентских запросов для формирования динамических страниц с соответствующим его обработчиком, среди которых наиболее популярным на сегодняшний день является язык PHP с соответствующим обработчиком скриптов на PHP [1, 12, 13]. Функциональным ядром этих систем с точки зрения формирования ответа на запрос является система управления контетом типа Joomla. Для организации хранения данных обязательным элементом является использование СУБД типа MySQL, PostgreSQL или MS SQL Server [2, 3, 9].

При рассмотрении задачи имитационного моделирования такой структуры Internet-систем возникают подчас вопросы к решению данной задачи в силу разной интерпретации ролей, которые играют указанные выше составляющие в этих систем.

В качестве примера рассмотрим процесс работы Intenet-системы на базе web-сервера Apache, языка программирования PHP и его обработчика, системы управления контентом Joomla, СУБД MySQL. И так, пользователь Internet производит запрос требуемой начальной страницы по заданному URL (Uniform Resource Locator). При этом этот запрос транс-

лируется требуемому web-серверу Apache, который извлекает требуемую начальную страницу из внешней памяти и передает ее на обработку обработчику PHP. Следует заметить, что существующие современные версии PHP версии 5 не требуют обязательного наличия расширения .php для своих скриптов, в результате чего запрашиваемая страница может не иметь расширения php, но независимо от этого, она все равно отправляется на обработку обработчику PHP [6,12].

Рассмотрим более подробно функциональность обработчика PHP, чтобы определить его роль в процессе формирования ответа Internet-системы на полученный запрос. Обработчик PHP версии 3 является интерпретатором [12]. Это значит что в процессе обработки PHP скрипта обработчик пошагово анализирует и тут же сразу выполняет инструкцию за инструкцией PHP скрипта на "лету". При этом, если встречаются циклические фрагменты исходного кода , то обработчику PHP приходится каждый раз снова анализировать и выполнять их. В результате такого подхода скорость формирования ответа на запрос в случае сложных PHP скриптов низкая. В последних версиях PHP 4 и 5 считается, что обработчик PHP теперь является не интерпретатором, а компилятором [12]. Однако это не совсем так. Рассмотрим понятия, что такое компилятор и что такое транслятор.

И так, транслятор это специального программа которая осуществляет преобразование программы, написанной на одном языке, в программу написанной на другом языке или в некоторое внутреннее рабочее представление, например в байт-код. Под термином компилятор часто понимают тот же транслятор, хотя для точности можно сказать, что компилятор в отличие от транслятора на выходе имеет конечный машинный код, готовый для исполнения. Транслятор же в отличие от компилятора выдает некоторый внутренний код, на основе которого строится в дальнейшем реализация программы. Например, .exe файлы не являются конечным машинным кодом. Их окончательная кодировка формируется используемой операционной системой, в рамках которой происходит их выполнение.

На основании вышеизложенного становиться ясно, что PHP 4 и 5 являются гибридными обработчиками, которые сочетают в себе как свойства интерпретатора, так и свойства транслятора. С одной стороны, формируемый так называемый "байт-код" в результате трансляции является внутренним представлением PHP скрипта в теле обработчика PHP, который можно использовать многократно в процессе работы PHP скрипта. Это очень выигрышно в случае циклического исполнения PHP скрипта в целом или отдельных его фрагментов. С другой стороны, полученный байт-код не сохраняется после завершение работы PHP скрипта. Это позволяет говорить, что по большому счету обработчики и PHP 4 и PHP 5 являются интерпретаторами, но более усовершенствованными, чем обработчик PHP 3.

Далее, после обработки начальной страницы PHP обработчиком происходит загрузка интегрированной среды функционирования системы управления Joomla, которая берет на себя ответственность по дальнейшему формированию ответа на запрос.

Систему управления контентом Joomla можно рассматривать как некоторое множество скриптов, написанных на языке PHP [2, 3, 9]. Система управления Joomla использует при формировании ответа такие составляющие как язык программирования Java-Script, документы в форматах HTML, XTML, XML, таблицы стилей CSS, которые позволяют делать сайт более интерактивными, использовать современную технологию Ajax, более эргономичными и привлекательными [2, 3, 9].

Таким образом, функционирование Joomla с учетом вышеотмеченного может быть в рамках только функционирования обработчика PHP, что позволяет с точки имитационного моделирования рассматривать работу этих составляющих в рамках единого процесса, без выделения работы Joomla в рамках отдельного процесса.

Чтобы определить место и роль следующей составляющей СУБД, следует обратить внимание на следующие особенности функционирования системы управления контентом Joomla. С точки зрения работы обработчика PHP, работа с Joomla представляет собой некоторую последовательность обработки PHP скриптов, которые выполняют те или иные действия. Следует отметить, что в процессе работы Joomla все необходимые данные, которые передаются от PHP скрипта к PHP скрипту, общие параметры окружения и прочие рабочие данные в оперативной памяти компьютера не сохраняются. Они сохраняются во внешней памяти, то есть в базе данных [2, 9]. Следует отменить, что в настоящее время Joomla поддерживает работу с такими СУБД как MySQL 5.1+, PostgreSQL 8.3.18 +, MS SQL 10.50.1600.1+ [9]. Наиболее популярной среди СУБД в Internet является СУБД MySQL [1,13]. Следует также отменить, что наиболее популярным web-сервером является web-сервер Apache [1, 6, 8, 13].

На основании вышеизложенного можно построить следующую общую схему Internet-систем на базе web-сервера Apache, PHP (Interpreter), системы управления контентом Joomla, СУБД MySQL с использованием принятых обозначений для имитационного моделирования [4, 5], которая представлена на рис. 1.

Остановимся кратко на основных моментах работы схемы модели имитационного моделирования. Генератор заявок генерирует заявки по экспоненциальному закону распределения, так как в большинстве случае такая интерпретация вполне уместна с практической точки зрения [4, 5]. Далее, входные заявки поступают на вход web-сервера Apache. Если все каналы web-сервера Apache заняты и входной буфер полностью заполнен входными заявками, то заявка не обработавшись покидает систему. Иначе, если все каналы заняты и в буфере есть место, то заявка ставиться в очередь на ожидание обработки. По мере освобождения каналов заявки выбираются из входного буфера и передаются на обработку освободившемуся каналу.

После обработки в рамках web-сервера Apache заявка или покидает систему, если не требуется участие для обработки системы управления контентом Joomla, или поступает на обработку в PHP-интерпретатор, который обрабатывает скрипты Joomla. На схеме PHP-интерпретатор + системы управления контентом Joomla помечены просто как PHP +

Joomla. Поскольку PHP-интерпретатор является отдельной составляющей каждого канала web-сервера Apache, а не является общим для всех каналов web-сервера Apache, входного буфера заявок для PHP-интерпретатора нет.

Далее, после обработки заявка поступает в СУБД MySQL. Аналогично схеме обработки заявок в web -сервере Apache, если все каналы СУБД MySQL заняты и входной буфер полностью заполнен входными заявками, то заявка не обработавшись или покидает систему или идет на обработку PHP-интерпретатору (на рис. 1 штриховая линия) и далее на обработку в web-сервер Apache, и покидает систему. Реализация того или иного пути покидания системы отклоненной заявки определяется степенью детализации имитационного моделирования . Иначе, если все каналы заняты и в буфере есть место, то заявка ставиться в очередь на ожидание обработки. По мере освобождения каналов заявки выбираются из входного буфера и передаются на обработку освободившемуся каналу. После обработки заявки в СУБД MySQL, она либо покидает систему (на рис. 1 штриховая линия) или идет на обработку PHP-интерпретатору и далее на обработку в web-сервер Apache и покидает систему. Аналогично сказанному выше, реализация того или иного пути покидания системы обработанной заявки определяется степенью детализации имитационного моделирования.

Для определения адекватных значений параметров элементов имитационной модели необходимо инсталлировать локальный web-портал, включая web-сервер Apache, PHP-интерпретатор, систему управления контентом Joomla, СУБД MySQL в рамках предполагаемого технического и программного обеспечения. Локальная инсталляция web-портала позволяет пренебречь временными затратами на передачу данных в процессе определения значений параметров моделирования. Одним из возможных путей определения значений параметров системы управления контентом Joomla и СУБД MySQL с точки зрения имитационного моделирования являются следующие действия.

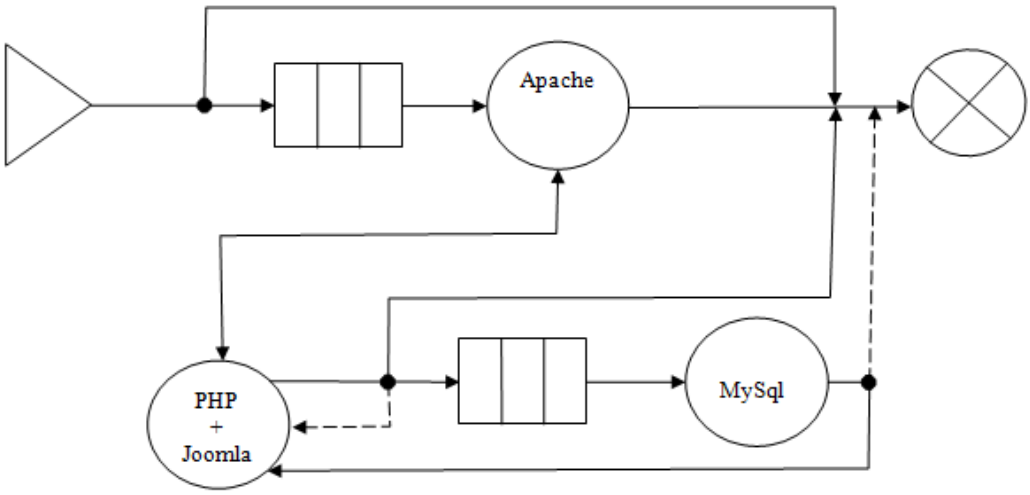


**Рисунок 1.** Схема имитационной модели

Необходимо войти в административную часть системы управления контентом Joomla. Перейти на вкладку "Система", затем выбрать "Общие настройки" на панели слева, далее выбрать вкладку "Система" и установить "Параметры отладки системы" в состояние "Да" [2, 9]. И затем выполнить действие по сохранению выбранного значения параметра. Тем самым, после этого, система управления контентом Joomla начинает включать в данные web-страницы Debug данные. В результате этого становиться доступной консоль отладки системы управления контентом Joomla на каждой странице web-портала. На вкладке "результаты профилирования" консоли отладки системы управления контентом Joomla становиться доступно время, потраченное системой управления контентом Joomla на формирование любой выбранной страницы web-портала, включая время, потраченное при этом на работу с СУБД MySQL.

Таким образом, получив эти данные для всех страниц web-портала, можно определить математическое ожидание времени работы PHP-интерпретатора совместно с системой управления контентом Joomla и математическое ожидание времени работы СУБД MySQL в рамках всего web-портала.

Для определения математического ожидания времени работы веб-сервера Apache (за вычетом времени работы PHP-интерпретатора совместно с Joomla и СУБД MySQL) можно поступить следующим образом.

Разработчики web-браузеров Google Chrome и Mozilla Firefox создали хорошие инструменты для web-разработчиков для отслеживания трассировки выполнения запросов к web-порталам (сайтам), которые позволяют определить тип и количество загружаемых файлов, их размеры, время их загрузки и интерпретации и другие параметры [7, 10]. В рамках данной статьи рассмотрим только "Инструменты разработчика" Google Chrome, одним из способов вызова которых в рамках операционной системы Windows является комбинация клавиш "Ctrl+Shift+I". "Инструменты разработки" Mozilla Firefox аналогичны вышеуказанным инструментам.

После перехода в режим "Инструменты разработчика", необходимо перейти на вкладку "Network" и пометить блок "Disable Cash", тем самым устраняя кэширование web-страниц из памяти компьютера при проведении экспериментальных действий. На этой вкладке будет необходима следующая информация: какие файлы загружались в процессе отображения страницы, и сколько времени потребовалось web-серверу Apache для формирования ответов на запросы. Для этих целей для каждого файла на вкладке "Timeline" (трассировочный график работы web-браузера Google Chrome) находим параметр "Waiting" (TTFB), который по сути, показывает время формирования ответа web-сервера Apache на запрос от web-браузера Google Chrome. Таким образом, суммируя значение параметр "Waiting" для всех переданных данных для определенной страницы мы получаем общее время реакции web-портала на полученный запрос по отображению этой web-страницы. Далее, поступая аналогичным образом для всех остальных страниц web-портала, можно определить математическое ожидание времени реакции web-портала на

запрос без учета временных затрат на передачу данных. Зная математическое ожидание времени отклика web-портала, которое включает математическое ожидание времени работы PHP- интерпретатора совместно с системой управления контентом Joomla и математическое ожидание времени работы СУБД MySQL, путем простого вычитания 2-х последних указанных значений из 1-го указанного значения можно получить математическое ожидание времени работы самого web-сервера Apache в рамках времени отклика web-портала в целом. Кроме того, если необходимо в рамках имитационного моделирования учесть, сколько времени при отклике web-портала тратится web-сервером Apache непосредственно на совместную работу с системой управления контентом Joomla и на работу без ее участия, анализируя содержимое 1-го загруженного файла можно определить по URL для каких в последующей загрузке файлов использовалась система управления контентом Joomla, а для каких нет [2, 7, 9]. Обработав эти полученные данные вышеуказанным способом можно получить математические ожидания работы web-сервера Apache непосредственно совместно с системой управления контентом Joomla и его работы без нее. Таким образом, мы получаем все необходимые значения данных для адекватного моделирования времени обслуживания web-сервера Apache, PHP- интерпретатора + системы управления контентом Joomla и СУБД MySQL, распределенного по экспоненциальному закону.

Далее для определения размеров буфера для входных запросов Apache и числа каналов (потоков) web-сервера Apache необходимо произвести следующее.

Web-сервера Apache может поддерживать несколько методов для организации многопоточной обработки запросов. Для определения того, на какой метод настроен web-сервер Apache необходимо запустить утилиту "httpd -V" в командной строке, которая выводит наименование текущего установленного "Multi-Processing Module" [6].

Рассмотрим это на примере операционной системы Windows. В этом случае этот будет модуль "MPM_WinNt". Для этого модуля размер входного буфера web-сервера Apache будет равен 511, а количество потоков (каналов) задается в настройках конфигурационного файла httpd-mpm в директории конфигурации web-сервера Apache [6]. По умолчанию это обычно 150.

Для определения значения параметров СУБД MySQL можно воспользоваться следующей процедурой действий. Для определения размера входного буфера в консоли СУБД MySQL необходимо запустить команды "select @@global.back_log;" или "show variables like "back_log"; Для операционной системы Windows это будет по умолчанию 80. Для определения числа каналов, можно запустить команду "select @@global.max_connections;" или "show variables like "max_connections";". По умолчанию это обычно 151. [11]

Таким образом, в результате вышеизложенного можно получить все необходимые исходные данные для построения программы для модели имитационного моделирования,

позволяющей провести адекватное имитационное исследование Internet-систем, построенных с использованием системы управления контентом Joomla.

## Список литературы


1. Афанасьев Г.И., Тимофеев В.Б. Использование систем управления контентом для разработки и сопровождения web-приложений в целях учебного процесса //Инженерный вестник. МГТУ им. Н.Э.Баумана. Электронный журнал. 2013. № 11. Режим доступа: http://engbul.bmstu.ru/doc/640819.html (дата обращения 14.03.2015)
2. Декстер М., Лэндри Л., Joomla programming: пер. с англ. М.: Вильямс. 2013 – 592с [M. Dexter, L. Landry. Joomla programming, Addison-Wesley, 2012. 594p .]
3. Мэрриотт Дж., Уоринг Э., Joomla 3.0. Официальное руководство: пер. с анг. -СПб: Питер. 2013-496с. [J.Marriot, E. Waring. The Official Joomla Book, 2nd ed., Addison-Wesley, 2012. 512p .]
4. Томашевский В.Н. Жданова Е.Г. Имитационное моделирование в среде GPSS.-М: Бестселлер, 2003.-416 с
5. Черненький В. М., Ревунков Г. И., Постников В. М. Эксплуатация АСОиУ : метод. указания к выполнению курсовой работы / Черненький В. М., Ревунков Г. И., Постников В. М. ; МГТУ им. Н. Э. Баумана. - М. : Изд-во МГТУ им. Н. Э. Баумана, 2009. - 28 с.
6. Apache official website , [Электронный ресурс]. Режим доступа: http://www.apache.org (дата обращения 14.03.2015)
7. Google Chrome developer official website, [Электронный ресурс]. Режим доступа https://developer.chrome.com/devtools/docs/network (date access 14.03.2015)
8. iTrack official website, [Электронный ресурс]. Режим доступа: http://itrack.ru/research/cmsrate/ (дата обращения 14.03.2015)
9. Joomla official website, [Electronic resource]. Access mode: http://www.joomla.org/ (date access 14.03.2015)
10. Mozilla addons official website, [Электронный ресурс]. Режим доступа: https://addons.mozilla.org/ru/firefox/addon/firebug/ http://www.joomla.org/ (date access 14.03.2015)
11. MySQL official website, [Электронный ресурс]. Режим доступа: http://www.mysql.com/ (дата обращения 14.03.2015)
12. PHP official website, [Электронный ресурс]. Режим доступа: http://www.php.net/ (дата обращения 14.03.2015)
13. Q-SuccessWorld company, Wide web technology surveys website, [Электронный ресурс]. Режим доступа: http://w3techs.com (дата обращения 14.03.2015)