

УДК 004.054

ИНТЕНСИФИКАЦИЯ НАГРУЗОЧНОГО ТЕСТИРОВАНИЯ С ПОМОЩЬЮ АППАРАТНОГО МОДУЛЯ, ПОСТРОЕННОГО НА БАЗЕ ПЛИС**Бородин А.А.***ФГБОУ ВПО «Московский государственный университет леса»,
Мытищи, e-mail: AntonioBorodin@gmail.com*

Надежное функционирование информационных систем является важнейшим фактором развития современной цивилизации. Это обуславливает актуальность и практическую значимость работ, направленных на решение данной задачи. Одним из основных способов ее решения является тестирование информационных систем. Под тестированием понимается исследование системы с целью проверки соответствия ее работы предъявляемым требованиям. В настоящий момент существуют различные виды тестирования, каждый из которых направлен на диагностику отдельной части системы. Для информационных систем глобальной сети важнейшим является нагрузочное тестирование. С помощью него удается измерить характеристики и проверить функционирование системы при эксплуатационных условиях. Улучшением данного процесса занимаются многие исследователи. Однако стадии запуска тестов и сбора результатов уделяется, на наш взгляд, недостаточно внимания. При выполнении данной стадии создается нагрузка на тестируемую систему и собираются результаты. Практика нагрузочного тестирования показывает, что достаточную величину нагрузки не удается создать с помощью одного узла. Поэтому применяются методы ее увеличения с помощью совокупности соединенных компьютеров. В данной статье представлены результаты испытания иного подхода, позволяющего интенсифицировать процесс создания нагрузки. Он основан на применении специализированного аппаратного блока, который построен на базе ПЛИС. Результаты испытания спроектированного прототипа, демонстрируют высокие нагрузочные способности.

Ключевые слова: нагрузочное тестирование, ПЛИС, запуск тестов, создание нагрузки, аппаратный нагрузчик

THE INTENSIFICATION OF LOAD TESTING BY USING FPGA-BASED HARDWARE**Borodin A.A.***Moscow State Forest University, Mytitschi, e-mail: AntonioBorodin@gmail.com*

Reliable functioning of information system is a key factor of civilization progress. This cause relevance and practical significance of works focused on solving that task. One of the main ways of solving this problem is a testing. Testing is a system research process with a goal to check compliance of its behavior to the shown requirements. At current time there are different types of testing. Each type is intended for specific part of system. For information systems of global network the most important type is a load testing. Load testing helps to estimate characteristics and to check functions of a system under the operation conditions. Many scientist works on improvement of that process. Nevertheless stage of the launch tests is getting less attention, to our opinion. At that stage loading is being created for a system and performance metrics are being gathered. Practice of load testing shows that enough amount loading cannot be created using only one computer. That is why there exist methods of increasing of loading by using connected computers. This article represents experimental results of other method. It is based on application of specialized hardware that is constructed with FPGA. Results show a high load capability of a created prototype.

Keywords: load testing, FPGA, test launching, load creation, hardware loader

В настоящее время информационные системы превратились из вспомогательных научных инструментов в центральные компоненты современной цивилизации. В постиндустриальном обществе, за редким исключением, жизнь каждого человека прямо или косвенно связана с ними. Управление, профессиональная деятельность, обучение, развлечение и т.д. уже немислимы без информационных систем, являющихся сосредоточением информации – ценнейшего ресурса нашей цивилизации. По этой причине к их надежности предъявляются чрезвычайно высокие требования.

Высокое качество работы информационных систем обеспечивается за счет тестирования. Тестирование представляет собой процесс исследования системы с целью

проверки соответствия ее работы предъявляемым требованиям. Существует большое количество разнообразных видов тестирования, охватывающих различные аспекты. Для информационных систем глобальной сети одним из наиболее важных видов тестирования является нагрузочное. Его целью является проверка функционирования и сбор характеристик в условиях, близких к эксплуатационным.

Нагрузочное тестирование является сложным процессом, включающим множество шагов. Усилия исследователей в настоящий момент сосредоточены на вопросах создания тестов [3–5, 9, 11, 12], разработки нагрузочных моделей [14], а также анализа полученных результатов [8, 15]. Есть работы, посвященные разработке методик на-

грузочного тестирования распределенных систем [2]. В то же самое время вопросам запуска тестов и сбора результатов уделяется недостаточное внимание. При их решении полагаются на программные средства для нагрузочного тестирования. Несмотря на их многообразие, все они обладают одинаковыми недостатками. В большинстве случаев при использовании одного компьютерного узла создать достаточную величину нагрузки не удается. Это обусловлено множеством аппаратных и программных факторов. Поэтому применяют совокупности соединенных компьютеров, представленных в виде распределенных, кластерных и облачных вычислений. Таким образом, нагрузочное тестирование информационных систем требует подключения большого количества компьютеров и дополнительных затрат. На преодоление этих недостатков и повышение интенсификации процесса направлена наша работа.

Стадия запуска тестов представляет собой процесс создания нагрузки на целевую систему и сбор метрик производительности на основе анализа ответов. При этом нагрузка представляет собой множество тестовых запросов, отправляемых с большой интенсивностью. Процесс отправки запроса можно разделить на 2 части:

- в соответствии с тестовыми скриптами осуществляется генерация запроса, который затем передается операционной системе;
- операционная система обеспечивает передачу и доставку запроса на тестируемую систему.

При этом время, необходимое операционной системе на обработку и передачу запроса, определяет предельную величину интенсивности, которую может достичь компьютерный узел. Таким образом, эта величина является нагрузочной способностью компьютерного узла. В нашей работе [1] мы провели эксперименты с целью измерения данной величины на популярных ОС. В ходе экспериментов была выявлена следующая особенность. На каждый тестовый запрос исследуемая система формирует ответ, который подвергается анализу. Ввиду того, что в ходе нагрузочного тестирования создается большая нагрузка, на нагружающую систему направляется большой поток ответов. Проведенные эксперименты позволили обнаружить, что этот поток ответов снижает нагружающие способности узла. В нашей работе мы назвали данную особенность эффектом обратной нагрузки.

Для интенсификации стадии запуска тестов и сбора результатов мы предлагаем иной подход. Он основан на применении специализированного аппаратного блока,

построенного на базе ПЛИС, который исполняет функции операционной системы по доставке сообщений. Он создает нагрузку и собирает метрики производительности. При этом подготовка тестовых запросов и параметров осуществляется с помощью компьютерного узла. После передачи этих данных на аппаратный нагрузчик он обеспечивает передачу запросов с требуемой интенсивностью и собирает необходимые метрики производительности.

На основе этой идеи был спроектирован прототип устройства, названный нами – аппаратный нагрузчик информационной системы (АНИС). Он был построен на основе платы разработки Terasic De2-115 с ППВМ ПЛИС Altera Cyclone IV [6]. На рис. 1 представлена обобщенная схема прототипа.

Ключевые компоненты, необходимые для генерации пакетов и обеспечения их доставки, реализованы в устройстве аппаратно. В состав АНИС также входит процессорная система, назначением которой является конфигурация параметров и обеспечение загрузки тестовых запросов. Ввиду того, что плата разработки содержит ограниченный набор памяти, собираемые метрики производительности выгружаются через отдельный интерфейс на компьютер. Для выгрузки данных АНИС содержит клиент базы данных Redis [13]. Таким образом, результаты нагрузочного тестирования оказываются в базе данных Redis, которая находится на отдельном компьютере. При разработке прототипа учитывались недостатки существующих систем. Так, обработкой исходящих и входящих данных занимаются независимые блоки, поэтому эффект обратной нагрузки на АНИС не проявляется. Ранее нами были проведены измерения предельных характеристик прототипа. Они показали высокие характеристики АНИС. Так, на формирование одного пакета с запросом устройству необходимо в среднем 220 нс. В данной статье представлены результаты испытания прототипа для создания нагрузки на веб-сервер.

Для испытания применялись узлы со следующими характеристиками:

- экспериментальная платформа (ЭП) № 1, компьютер Intel Core i7 920 2,67 ГГц, DDR3-1066 9ГБ, CentOS 6.4;
- экспериментальная платформа № 2, ноутбук Samsung R580-JS03 Intel Core i5 430 М 2,26 ГГц, DDR3-1066 3 ГБ, Ubuntu 12.10;
- экспериментальная платформа № 3, ноутбук MacBook Air Intel Core i7 3667U 2 ГГц, DDR3L-1600 МГц 8 ГБ, OS X 10.8.5.

В качестве нагружаемых систем выступали экспериментальные платформы 1 и 2 с установленным веб-сервером NGINX

[10]. После установки веб-сервера, настройки и дополнительные модули были оставлены без изменений. В ходе ответа на запрос он формировал стандартную html страницу. Результаты работы прототипа сопоставлялись с результатами широко используемого

в настоящее время инструмента нагрузочного тестирования Apache JMeter [7]. Источником информации о созданной нагрузке служат лог файлы сервера, в которых регистрируется каждый поступивший запрос и его время.

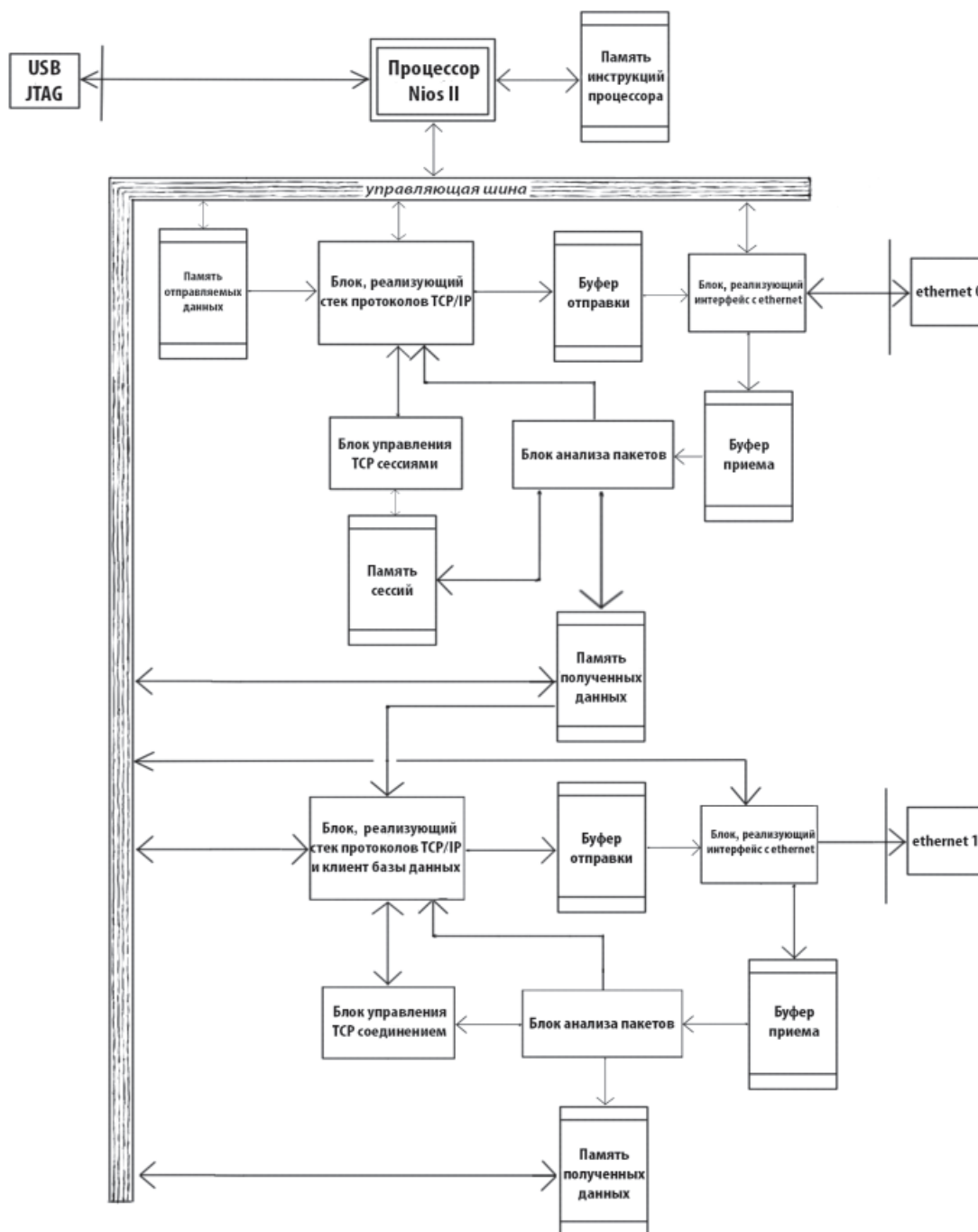


Рис. 1. Структура разработанного прототипа

Инструмент нагрузочного тестирования запускался сначала на одной экспериментальной платформе, а затем на другой. Кроме того, параметры программы Jmeter на

каждой платформе определялись эмпирическим путем так, чтобы создаваемая нагрузка была наибольшей. Результаты испытаний представлены на графиках (рис. 2 и 3).

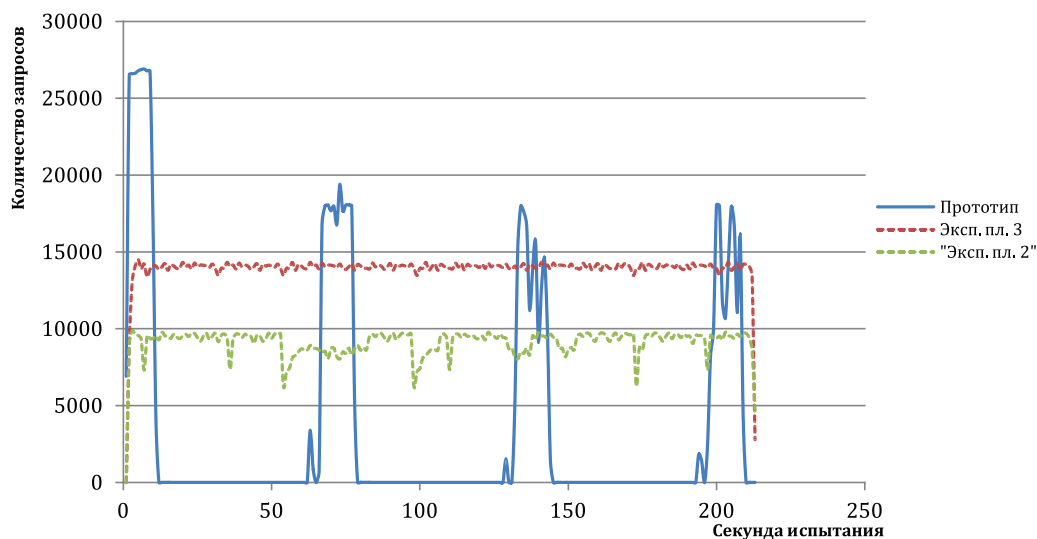


Рис. 2. Испытание прототипа аппаратного нагрузчика на экспериментальной платформе № 1

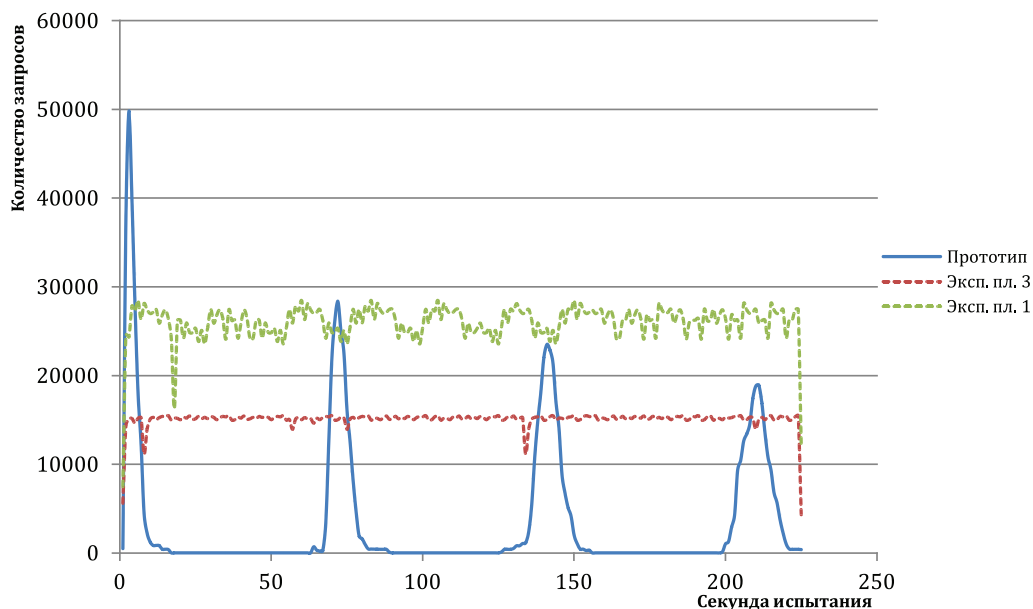


Рис. 3. Испытание прототипа аппаратного нагрузчика на экспериментальной платформе № 2

Представленные графики демонстрируют, что опытный прототип в ходе испытаний создавал кратковременную значительную нагрузку с некоторой периодичностью. Проведенный анализ данных, полученных с помощью перехватчиков пакетов, показал, что это связано с высокой интенсивностью генерации запросов прототипом. Во время создания нагрузки устройство отправляло запросы веб-серверу и ожидало ответа от него в пределах сессии. Как показывают данные перехватчика пакетов, сервер получал запрос, но не всегда отвечал

на него. Ввиду того, что в составе текущей версии прототипа отсутствуют механизмы таймеров, устройство бесконечно долго ожидало ответа от сервера. Через некоторый промежуток времени, определяемый настройками веб-сервера и операционной системы, информационная система разрывала соединение с прототипом, полагая, что запросов с его стороны не поступило. Получив запрос на разъединение, аппаратный нагрузчик перезапускал соединение и продолжал отправлять запросы. По этой причине временные промежутки между

всплесками примерно одинаковы. Кроме того, как видно на графиках, пиковое значение интенсивности с каждым всплеском уменьшалось. Причиной этого явился тот же эффект, только в этом случае сама операционная система не отвечает на запрос открытия TCP сессии.

Проведенные испытания продемонстрировали практически возможности прототипа АНИС. Даже несмотря на недостатки текущей версии, полученные результаты показывают, как с помощью аппаратного модуля с ПЛИС возможно интенсифицировать процесс создания нагрузки и сбора результатов. Таким образом, прототип способен стать полноценным инструментом для задач нагрузочного тестирования.

Список литературы

1. Бородин А.А. Исследование нагрузочных способностей компьютерных систем // Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление. – 2014. – № 3(198). – С. 19–27.
2. Ермакин Алексей Александрович Разработка метода построения комплекса нагрузочного тестирования распределенной информационной системы: дис. ... канд. тех. наук: 05.13.13. – СПб., 2005. – 147 л.
3. Силаков Д.В. Автоматизация тестирования web-приложений, основанных на скриптовых языках // Труды ИСП РАН. – 2008. – № 2. – С. 159–178.
4. Сортвов А., Хорошилов А. Функциональное тестирование Web-приложений на основе технологии UniTesK // Труды Института системного программирования РАН. – 2004. – № 8. – С. 77–97.
5. Яковенко П.Н., Сапожников А.В. Инфраструктура тестирования веб-сервисов на базе технологии TTCN-3 и платформы .NET // Труды ИСП РАН. – 2009. – С. 63–74.
6. Altera DE2-115 Development and Education Board [Электронный ресурс]. – terasic, 2013. – Режим доступа: <http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=165&No=502>, свободный. – Загл. с экрана.
7. Apache JMeter [Электронный ресурс]. – The Apache Software Foundation Apache JMeter, 2013. – Режим доступа: <http://jmeter.apache.org>, свободный. – Загл. с экрана.
8. Haroon Malik, A Methodology to Support Load Test Analysis // ICSE '10 Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering. – Vol. 2. – P. 421–424.
9. Kevin Morrison, Hisham M. Haddad Converting users to testers: an alternative approach to load test script creation, parameterization and data correlation // Journal of Computing Sciences in Colleges. – Vol. 28 Issue 2, December 2012. – P. 188–196.
10. nginx [Электронный ресурс]. – NGINX, 2013. – Режим доступа: <http://nginx.org/ru/>, свободный. – Загл. с экрана.
11. Pingyu Zhang, Sebastian Elbaum, Matthew B. Dwyer Automatic Generation of Load Tests // ASE '11 Proceedings of the 2011 26th IEEE/ACM International Conference on Automated Software Engineering. – P. 43–52.
12. Pingyu Zhang, Sebastian Elbaum, Matthew B. Dwyer Compositional load test generation for software pipelines // ISSTA 2012 Proceedings of the 2012 International Symposium on Software Testing and Analysis, P. 89–99.
13. Redis [Электронный ресурс]. – redis, 2013. – Режим доступа: <http://redis.io>, свободный. – Загл. с экрана.
14. Yuhong Cai, John Grundy, John Hosking Synthesizing client load models for performance engineering via web crawling // ASE '07 Proceedings of the twenty-second IEEE/ACM international conference on Automated software engineering. – P. 353–362.
15. Zhen Ming Jiang, Automated analysis of load testing results, ISSTA '10 Proceedings of the 19th international symposium on Software testing and analysis, pp. 143–146.

References

1. Borodin A.A. Issledovanie nagruzochnyh sposobnostej komp'yuternyh sistem Nauchno-tehnicheskie vedomosti SPbGPU. Informatika. Telekommunikacii. Upravlenie., 2014, no. 3(198), pp. 19–27.
2. Ermykin Aleksej Aleksandrovich Razrabotka metoda postroenija kompleksa nagruzochnogo testirovanija raspredelennoj informacionnoj sistemy: dis. ... kand. teh. nauk: 05.13.13., SPb., 2005, 147 p.
3. Silakov D.V. Avtomatizacija testirovanija web-prilozhenij, osnovannyh na skriptovyh jazykah, Trudy ISP RAN, 2008, no. 2, pp. 159–178.
4. Sortov A., Horoshilov A. Funkcional'noe testirovanie Web-prilozhenij na osnove tehnologii UniTesK, Trudy Instituta sistemnogo programirovanija RAN, no. 8, 2004, pp. 77–97.
5. Jakovenko P. N., Sapozhnikov A. V. INFRASTRUKTURA TESTIROVANJA VEB-SERVISOV NA BAZE TEHNOLOGII TTCN-3 I PLATFORMY.NET, Trudy ISP RAN, 2009, pp. 63–74.
6. Altera DE2-115 Development and Education Board, Available at: <http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=165&No=502> (accessed 15 October 2014).
7. Apache JMeter, available at: <http://jmeter.apache.org> (accessed 15 October 2014).
8. Haroon Malik, A Methodology to Support Load Test Analysis, ICSE '10 Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering - Volume 2, pp. 421–424.
9. Kevin Morrison, Hisham M. Haddad Converting users to testers: an alternative approach to load test script creation, parameterization and data correlation, Journal of Computing Sciences in Colleges Volume 28 Issue 2, December 2012, pp. 188–196.
10. Nginx, Available at: <http://nginx.org/ru/> (accessed 15 October 2014).
11. Pingyu Zhang, Sebastian Elbaum, Matthew B. Dwyer Automatic Generation of Load Tests, ASE '11 Proceedings of the 2011 26th IEEE/ACM International Conference on Automated Software Engineering, pp. 43–52.
12. Pingyu Zhang, Sebastian Elbaum, Matthew B. Dwyer Compositional load test generation for software pipelines, ISSTA 2012 Proceedings of the 2012 International Symposium on Software Testing and Analysis, pp. 89–99.
13. Redis, Available at: <http://redis.io> (accessed 15 October 2014).
14. Yuhong Cai, John Grundy, John Hosking Synthesizing client load models for performance engineering via web crawling, ASE '07 Proceedings of the twenty-second IEEE/ACM international conference on Automated software engineering, pp. 353–362.
15. Zhen Ming Jiang, Automated analysis of load testing results, ISSTA '10 Proceedings of the 19th international symposium on Software testing and analysis, pp. 143–146.

Рецензенты:

Корольков А.В., д.ф.-м.н., профессор, декан ФЭСТ, ФГБОУ ВПО «Московский государственный университет леса», г. Мытищи-5; Финогеев А.Г., д.т.н., профессор кафедры «Системы автоматизации проектирования», ФГБОУ ВПО «Пензенский государственный университет», г. Пенза.

Работа поступила в редакцию 28.11.2014.