

УДК 004.942

## РАЗРАБОТКА ВЫСОКОНАГРУЖЕННЫХ ПРИЛОЖЕНИЙ ДЛЯ СОЦИАЛЬНОЙ СЕТИ

*Плотников Д.Ю., Малёванный Е.Ф., Аноприенко А.Я.  
Донецкий национальный технический университет  
Кафедра компьютерной инженерии  
E-mail: dayko1@ukr.net*

*В статье исследуется вопрос выбора современных средств разработки, тестирования и анализа производительности веб-приложений, ориентированных на работу в социальных сетях.*

**Постановка задач исследования.** *Сегодня социальные сети становятся еще более популярными, чем раньше. Сейчас, так называемая «страна Facebook», уже насчитывает почти 1 миллиард человек. Люди не ограничиваются только общением на сайте, а активно используют приложения, размещенные на сайте. Приложения на сайте может разместить любой человек. Если идея интересная – приложением начинает пользоваться огромное количество человек, которое стремится к миллиарду. Поэтому способы разработки, тестирования, анализа приложений с высокой нагрузкой являются актуальными.*

### **Решение задачи и результаты исследований.**

В данном докладе мы рассмотрим пример построения высоконагруженного приложения для социальной сети. Мы затронем все этапы создания, кроме первого, на наш взгляд, решающего – поиска идеи. В нашем случае 2 приложения: голосование за самого красивого человека приложения и ответы на анонимные вопросы. После выбора идеи необходимо было найти сервер для тестирования, мы использовали 8-ядерный Intel Core i7-920 с 24 ГБ ОЗУ.

В качестве языка программирования был выбран PHP. В сети много споров по поводу того, какой язык быстрее работает. Существует 2 группы языков «серверного» программирования. В языках, которые относятся к первой группе, например в PHP при запуске скрипта он вначале каждый раз компилируется, а затем результаты выводятся на экран. Код, написанный на языках второго типа, компилируется 1 раз и при запросе страницы сразу же выполняется программа и выводятся результаты на экран. Естественно, на первый взгляд это быстрее чем в первом способе. Но не так всё просто. В качестве теста скорости была написана программа обработки массивов. На рисунке 1 изображен график скорости работы кода, одинакового по смыслу, но реализованного на различных серверных языках. Пример кода на языке C++ :

```
#include <iostream>
using namespace std;
int main(void) {
    long r = 0;
    for (int i = 0; i < 10000; i++) {
```

```

    for (int j = 0; j < 10000; j++) {
        r = (r + (i * j) % 100) % 47;
    }
}
cout << «answer: » << r << endl;
}

```

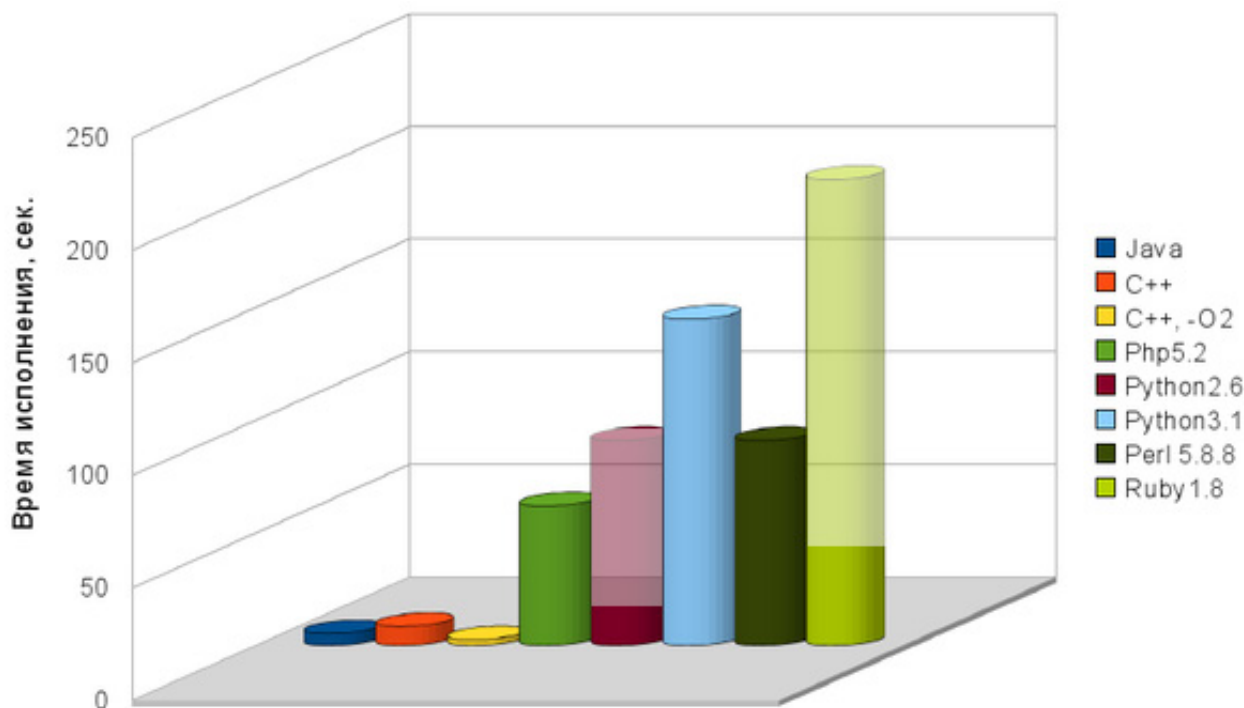


Рисунок 1. Скорость работы программы на различных серверных языках

Мы видим, что C++ работает во много раз быстрее других языков, но как оказалось – только если есть работа с большими данными и вычислениями. В наших же приложениях мы в основном используем только запросы к базе данных и формирование ответа, в виде JSON массива. Мы провели тестирование и выяснили, что результаты работы программы на PHP и C++ одинаковы, так как идут запросы к базе данных MySQL, и ограничивающим фактором является время отклика сервера с базой.

Программировать на PHP гораздо проще, удобнее и быстрее, чем на C++. Исходя из этого можно сделать вывод: часть файлов, где нужны запросы к базе данных – нужно реализовывать на более легком PHP, файлы, где только работы с данными – на C++.

Следующим этапом в разработке приложений является тестирование и анализ. Для тестирования мы использовали программу `ab` из `apache2-utils` и сайт `loadimpact.com`. Эта программа, на наш взгляд, идеальная для оценки эффективности работы приложения. В качестве входных параметров мы передаём количество общих запросов (ключ – `n`), количество одновременных запросов (ключ – `c`) и адрес файла. На рис. 2 приведен скриншот работы программы `ab` при тестировании файла, который соединяется с базой данных и формирует JSON массив данных, с информацией о пользователе, программа была запущена со следующими параметрами:

```
ab -n 1000000 -c 100 http://127.0.0.1/api/getinfo.php?id=1
```

Данные, которые получены с помощью программы `ab` ценны тем, что мы видим,

```

Server Software:      nginx/1.2.3
Server Hostname:     127.0.0.1
Server Port:         80

Document Path:       /api/getinfo.php?id=1
Document Length:     547 bytes

Concurrency Level:   100
Time taken for tests: 90.426 seconds
Complete requests:   1000000
Failed requests:     0
Write errors:        0
Total transferred:   710000000 bytes
HTML transferred:   547000000 bytes
Requests per second: 11058.73 [#/sec] (mean)
Time per request:    9.043 [ms] (mean)
Time per request:    0.090 [ms] (mean, across all concurrent requests)
Transfer rate:       7667.68 [Kbytes/sec] received

```

Рисунок 2. Тест скорости работы php-скрипта

сколько запросов в секунду может обработать сервер (Requests per second) и меняя код сразу видим изменение производительности. Таким образом, происходит оптимизация кода: изменили, сохранили и протестировали, записали количество запросов в секунду и Time per request. Далее еще раз изменили и протестировали. После десятка вариантов выбираем с минимальным Time per request и оставляем его.

На рис. 3, 4 изображены скриншоты программ в социальных сетях: «Милашка недели» и «Узнай меня» соответственно.



Рисунок 3. Приложение «Милашка недели»



Рисунок 4. Приложение «Узнай меня»

Для анализа работы приложения, уже после запуска проекта очень удобно пользоваться сервисом Яндекс.метрика. Он позволяет получать различные данные по пользователям приложения, такие как: страна, город, пол, возраст, системные данные, строить графики посещаемости и нагрузки и использовать инструмент «Вебвизор».

Вебвизор от Яндекса – настоящий прорыв в мире анализа работы веб-приложения. Вебвизор записывает все действия пользователя на сайте и мы в специальном плеере можем просмотреть движения курсора пользователя и его передвижение по сайту. Очень удобно оценивать навигацию по сайту и переходы между страницами. На рис. 5 изображена работа вебвизора, а на рис. 6, 7 – график посещений разработанных проектов.



Рисунок 5. Работа вебвизора



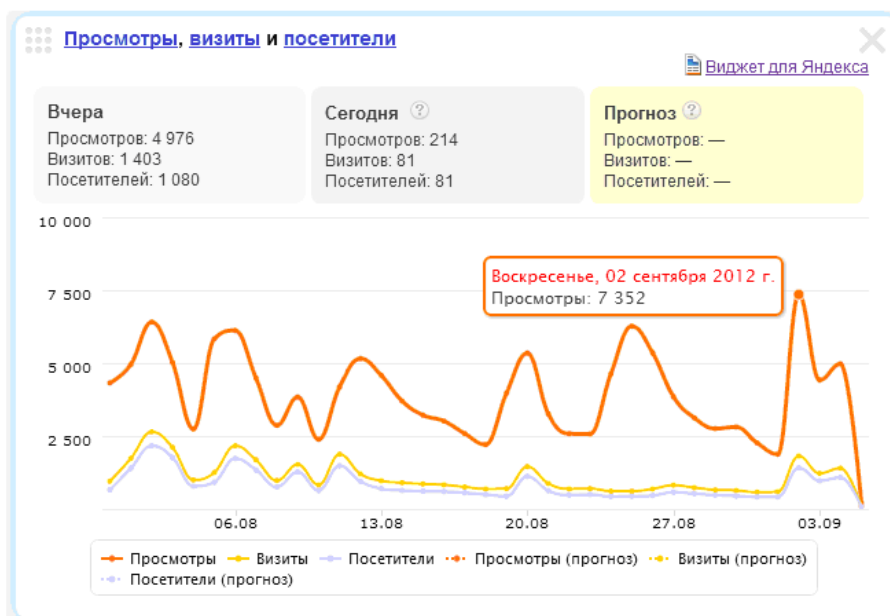


Рисунок 6. Нагрузка на приложение «Милашка недели»

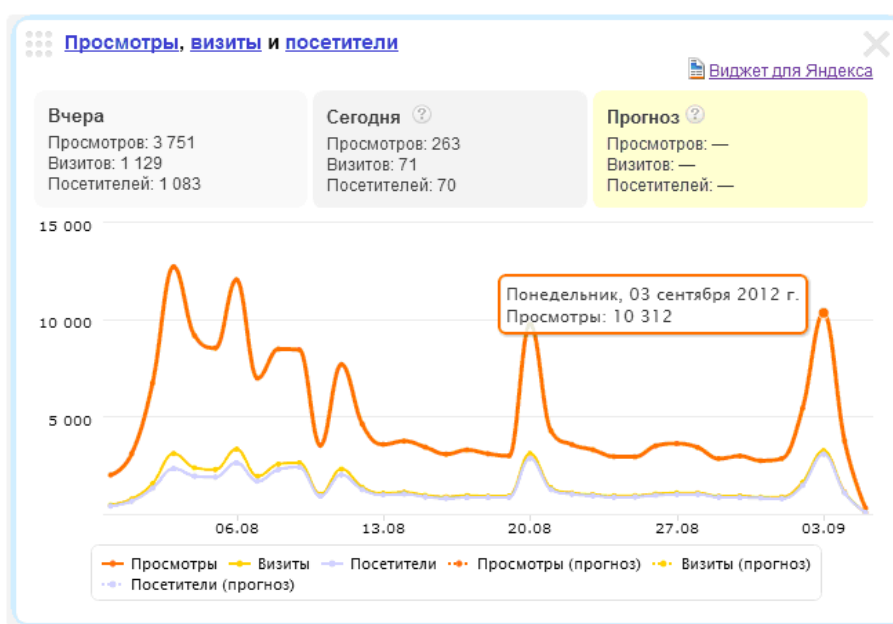


Рисунок 7. Нагрузка на приложение «Узнай меня недели»

## Выводы

В ходе разработки было создано 2 веб-приложения: «Милашка недели» и «Узнай меня». Были сформулированы правила для выбора языков программирования для различных задач и найдены инструменты для тестирования производительности и удобства приложения. Сейчас максимальная нагрузка была 127 пользователей в секунду и сервер с ней успешно справляется. Общее количество пользователей: 56727.

## Перечень источников

- [1] Дмитрий Котеров, Алексей Костарев PHP. В подлиннике. — Спб.: «БХВ-Петербург», 2005. — С. 1120. — ISBN 5-94157-245-X
- [2] Костарев А. Ф. PHP 5. — Спб.: «БХВ-Петербург», 2008. — С. 1104. — ISBN 978-5-9775-0315-0