

УДК 004.43

## ИССЛЕДОВАНИЕ ПОДХОДОВ И ИНСТРУМЕНТОВ ДЛЯ СОЗДАНИЯ СЕРВЕРНОЙ ЧАСТИ СИСТЕМ ТЕМАТИЧЕСКИХ СОЦИАЛЬНЫХ СЕТЕЙ

Гончар В.А. (Национальный исследовательский университет ИТМО)

Научный руководитель – доцент, кандидат педагогических наук Авксеньева Е.Ю.  
(Национальный исследовательский университет ИТМО)

В этой работе рассматриваются существующие серверные языки программирования для определения наиболее эффективного решения в разработке тематических социальных сетей. Для определения языка рассматриваются параметры поток/процесс, неблокирующий I/O, типизация и производительность. Определение эффективного языка позволит сократить издержки на использование оборудования и позволит увеличить максимально возможную аудиторию систем.

**Введение.** Социальная сеть – сложная система, обладающая большим количеством программных средств. Хотя в каждой социальной сети в теории одновременно не должно присутствовать большое количество пользователей, архитектура таких сетей может быть сложной и должна выдерживать потенциально большие нагрузки. Основной проблемой, где эту нагрузка может создавать проблемы, как в экономическом плане, так и в плане жизнеспособности программного обеспечения – является серверная часть программного обеспечения. На данный момент существует большое количество исследований для высоконагруженных систем, например, Кулькин Д.М. в статье «Выбор языка программирования для разработки серверной части веб-приложения» рассматривает выбор языка в зависимости от критериев гибкости и производительности, однако серверные языки развиваются, и стоит пересматривать их производительность для применения в современных системах. Также есть необходимость рассматривать серверные решения для конкретной задачи в отдельности, так как не все решения могут быть универсально применены к любой системе. В этой работе необходимо рассмотреть подходы к созданию серверной части для применения в тематических социальных сетях, а также определить наиболее оптимальные инструменты.

**Основная часть.** В качестве предлагаемых решений, стоит рассмотреть существующие серверные языки программирования, их критерии, для определения наиболее эффективного. Существует множество языков для написания кода для этой части систем. Можно привести следующие факторы, которые зависят от выбранного языка программирования: целевая платформа, гибкость языка, производительность, поддержка и сообщество.

Выполнение всех факторов приведет к уменьшению стоимости затрат разработки приложения и увеличению производительности, а как следствие – к увеличению аудитории и прибыли.

В качестве параметров для выявления эффективности языка, необходимо рассматривать наличие следующих подходов: поток/процесс, неблокирующий I/O, типизация, производительность, простота использования.

Для создания серверной части приложения обычно используются следующие языки программирования как самые популярные: NodeJS, PHP, C#, C++, Java, Python.

В чистом виде языки используются редко, в особенности если необходимо быстро разработать систему для первого запуска. Последующие изменения языков могут быть произведены в зависимости от потребляемых ресурсов и количества аудитории. Поэтому для сравнения имеет смысл взять NodeJS (Express, koa), Python (Django, Flask) и PHP (Laravel, Lumen, Yii2, Symfony, Zend), поскольку они более просты в изучении и обладают большим количеством фреймворков. Решения на языках Java, C++ и C# стоит рассматривать для последующего расширения социальных сетей на большие аудитории.

Подход с использованием потоков в веб-программировании, как правило, рассматривается для оптимизации и масштабирования систем. В небольших системах их использование неэффективно ввиду усложнения разработки. Социальная сеть – сложная с точки зрения

разработки и, как правило, высоконагруженная. Эти параметры стоит рассматривать в первую очередь при выборе языка программирования. Для всех перечисленных языков неблокирующий I/O присутствует либо в качестве оригинального решения для языка, либо в виде отдельной библиотеки.

Строгая типизация присуща для языков C++ и C#, этот параметр влияет на производительность, поэтому более важен.

Параметр «сообщество и поддержка» важен для более быстрой разработки решений, особенно если они сложные. Чем больше сообщество, тем больше существует уже готовых решений в виде пакетов и плагинов, и уже готовых решений, где можно посмотреть варианты реализации. Все рассматриваемые языки имеют достаточно активные сообщества.

Параметр производительности наиболее важен, так как более производительная система будет потреблять меньше ресурсов. Для исследования производительности можно обратиться к источнику «Web Framework Benchmarks», в котором приведены метрики для различных языков программирования и фреймворков. Основными метриками для теста производительности языков и фреймворков являются: JSON-сериализация, запрос к базе данных, множественный запрос к базе данных, выполнение кэшированных запросов, выбор произвольного количества записей через ORM, обновление данных в базе данных (1, 5, 10, 15 и 20 выборок на запрос), вывод обычного текста. Средневзвешенное значение этих параметров составляет совокупный балл. Из приведенной статистики можно получить следующую информацию: фреймворки на языке C++ имеют наибольшую производительность (Drogon – 9392 балла), фреймворки на C# имеют высокую производительность (ASP.NET – 7136 баллов), фреймворки на языке Java имеют высокую производительность, хотя разброс достаточно высокий (Vert.x – 6382 балла), фреймворки на PHP имеют плохие результаты (75-1400 баллов), однако если использовать библиотеку для неблокирующего кода Swoole, то результаты достаточно высокие (Swoole – 3736 баллов), NodeJS не показывает хороший результат – 1434 балла, фреймворки также набирают небольшое количество баллов (express – 620 баллов), решения на Python имеют плохие результаты (flask – 1213 баллов). Запущенные тесты на собственном оборудовании (Intel Core i7-6700k 4GHz, 64 GB RAM, SSD NVMe) подтверждают вышеперечисленные результаты производительности.

Исходя из полученных результатов производительности, можно сделать следующие выводы: языки и фреймворки применяющие неблокирующий I/O и строгую типизацию, наиболее эффективны; использование фреймворков замедляет производительность; наиболее популярные решения являются более непроизводительными. Решения на PHP с применением Swoole могут быть рассмотрены как конкурентные для достижения MVP.

**Выводы.** С учетом сделанных выводов по результатам производительности, в качестве основных решений для создания серверной части приложений тематических социальных сетей, можно использовать языки с неблокирующим I/O и строгой типизацией, в частности: C++ (с применением фреймворка Drogon) или C# (с применением фреймворка ASP.NET) – их стоит рассматривать в том случае, если аудитория социальных сетей будет возрастать. Для реализации MVP следует использовать PHP с применением неблокирующего I/O (с применением Symfony и фреймворка Swoole), поскольку это решение позволит быстрее реализовать системы за счет большего сообщества и написания меньшего количества кода. Применение вышеописанных подходов и технологий позволит сократить издержки на использование оборудования и увеличить максимально возможную аудиторию, которую смогут обслужить разработанные системы.

Гончар В.А. (автор)

Авксеньева Е.Ю. (научный руководитель)