

Селькин Виталий Евгеньевич

магистрант

ФГАОУ ВО «Санкт-Петербургский национальный
исследовательский университет информационных

технологий, механики и оптики»

г. Санкт-Петербург

ВРЕМЕННОЙ АНАЛИЗ МОДУЛЬНОЙ СБОРКИ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА НА JAVASCRIPT

Аннотация: в данной статье производится оценка эффективности модульного принципа на предмет временной задержки, которой обладают многосоставные приложения. Результат анализа исследуемой характеристики позволит оценить степень отрицательного влияния модульной архитектуры на этапе сборки в современных приложениях.

Ключевые слова: модульная архитектура, пользовательский интерфейс, JavaScript, временная задержка.

Пользовательский интерфейс является одним из ключевых элементов в архитектуре многосоставных приложений, обладающих сложной программной логикой. Основной причиной, свидетельствующей о несостоятельности использования стандартных решений в реализации интерфейса подобных приложений, является его сложный многоэлементный характер. Решением данной проблемы может служить метод разработки, при котором отдельные элементы рассматриваются как самостоятельные части, обладающие собственным внешним видом и поведением [1]. Принцип, реализующий подобную возможность, называется модульным [2].

Основным препятствием на пути модульной организации лежит стандартная транспортная модель, представляющая каждый ответ сервера как отдельно сформированную страницу. Поэтому для придания многоэлементного характера пользовательскому интерфейсу в этап его разработки вводят сборочную фазу,

осуществляющую построение всех его зависимостей. Главным недостатком подобных систем является временная задержка данной фазы. В рамках текущей статьи предлагается оценить степень её влияния на примере проведённого эксперимента с тестовой системой.

Метод тестирования заключался в многократном запуске данного процесса и замере временной метки с момента его запуска. В целях исключения влияния системной конфигурации все запуски были осуществлены в рамках одного рабочего окружения, под управлением компьютера со следующей конфигурацией (см. табл. 1):

Таблица 1

Конфигурация компьютера для запуска системы

CPU	Intel Core i5–650, 3.2GHz
RAM	8 Gb
ОС	Arch Linux, x86–64
Платформа	io.js 3.3.1

Для проведения опыта было использовано 20 итераций. Тестовый пользовательский интерфейс приложения включал 19 модульных элементов, основанных на JavaScript. Исследуемый процесс содержал только программную реализацию [3] прототипа пользовательского интерфейса и не учитывал его визуального оформления.

Кроме того, в целях исключения влияния сторонних факторов, в исследовании не использовались инструменты синтаксического преобразования. Так как все модули реализуют систему импорта, то стоит отметить, что глубина вложенности не превышала 6 элементов (3 ветки с глубиной по 6 модулей и один корневой). Запуск производился в рамках одного процесса в потоковом режиме обработки. Графическая среда для операций вывода не использовалась, все данные выводились на терминал в консольном режиме.

Стоит также отметить, на выходе запускаемого процесса все зависимости были упакованы в один единый модуль, чтобы часть процессорного времени не расходовалась на проверку подобия модулей.

По результатам проделанного опыта были получены значения, выраженные в секундах, которые, в последствие, были разделены на 4 интервала. На основании данного разделения была построена гистограмма попадания количества запусков во временной интервал задержки (рис. 1).

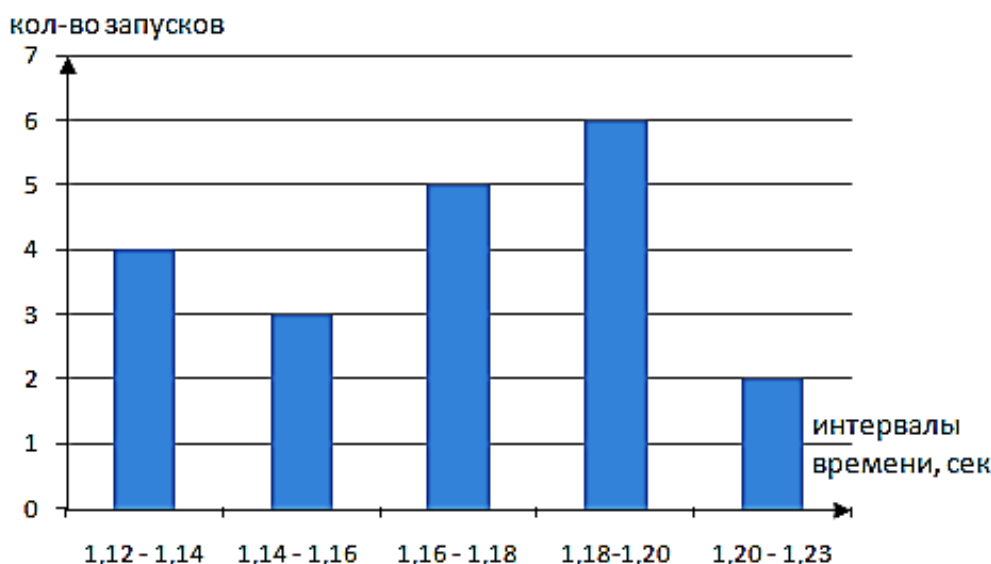


Рис. 1. Гистограмма попадания кол-ва запусков сборки во временной интервал задержки

Из представленного результата видно, что максимальная задержка составила 1230 мс. При этом большая часть запусков попала в интервал от 1,16 до 1,2 с. Учитывая тот факт, что многосоставные приложения имеют длительный цикл разработки, можно сделать вывод, что влияние временной задержки сборочной фазы оказывает значительно малый эффект на временной цикл по сравнению с преимуществами модульной архитектуры.

Данная работа проводилась в рамках исследовательской работы по анализу модульного принципа. Перспективным направлением развития является анализ временных характеристик, отражающих модульность стилевых элементов.

Список литературы

1. The standart HTML Imports // W3C.GITHUB.IO: официальный репозиторий со списков рабочих проектов World Wide Web Консорциума, 2016 [Электронный ресурс]. – Режим доступа: <http://w3c.github.io/webcomponents/spec/imports/> (дата обращения: 01.04.2016).

2. Buckett C. Web Components in Action. – Manning Publications Company, 2015. – 225 с.

3. The specification ECMA-262 // WWW.ECMA-INTERNATIONAL.ORG: официальный сайт ассоциации Ecma International, 2016 [Электронный ресурс]. – Режим доступа: <http://www.ecma-international.org/publications/standards/Ecma-262.htm> (дата обращения: 01.04.2016).