

В данной работе описываются анализа инструментов визуализации и создания оптимального алгоритма визуализации данных для сравнения производительности двух произвольных различных функций языка JavaScript. Анализ осуществляется путем выявления основных критериев для оценки оптимальности визуализации и проведения измерений в среде браузера с помощью встроенного модуля «Performance».

Введение. С началом пандемии и переходом многих организаций и учебных заведений на удалённую работу и обучение многие веб-приложения претерпели чрезвычайно большую нагрузку. Это сказалось на быстродействии и производительности их работы. Оптимизация веб-приложений и по сей день остаётся одной из важнейших задач для разработчиков в этой области. Визуализация – один из самых эффективных методов отображения информации. Для систем сравнения производительности и затрат памяти функций языка программирования визуализация имеет ключевое значение: она позволяет разработчику понять наглядно, в какой ситуации, при каком объёме данных какую функцию было бы лучше использовать для повышения производительности программного кода. Анализ и исследование инструментов визуализации позволит создать оптимальный алгоритм сравнения производительности двух произвольных различных функций языка JavaScript, способствующий оптимизации визуализирования данных, улучшению и ускорению процесса обработки данных, полученных при сравнении, путём внедрения данного алгоритма в рабочую систему. В качестве инструментов для визуализации были выбраны следующие: *D3.js*, *amCharts*, *AnyChart*, *Chart.js*, *Google Charts*, *Plotly.js* и *ZingChart*.

Основная часть. Основными критериями для оценки оптимальности визуализации были выбраны: объем памяти, занимаемый средством визуализации; скорость обработки данных и их визуализации; тип данных, принимаемый на вход функцией визуализации.

Для определения объема занимаемой памяти были проведены расчёты на основании объема исходных файлов используемых средств визуализации с учётом лишь тех файлов и директорий, которые необходимы для полного функционирования библиотеки. При использовании *Google Charts* вычисления производятся на облачном сервере, поэтому для него расчёт проведён не был.

Анализ поддерживаемых форматов импорта выявил, что все рассматриваемые инструменты визуализации поддерживают наиболее распространённые и оптимальные наборы данных (CSV, JSON, XML), следовательно, сравнение по весу исходных данных не производилось. Однако в рамках данного исследования формат CSV являлся наиболее приоритетным по причине своей легковесности: файл этого формата он весит в 2 раза меньше, чем файл JSON с аналогичными данными и в 2.3 раза меньше, чем файл XML.

В качестве источника данных для сравнения скорости обработки данных и визуализации был использовал датасет *Minimum Daily Temperatures Dataset*. Этот датасет содержит сведения о минимальных дневных температурах за 10 лет с 1981 по 1990 год в городе Мельбурн и состоит из 3650 объектов.

Измерение времени выполнения визуализации данных было произведено с помощью встроенного модуля «Performance». Визуализация данных осуществлялась в среде браузера, поэтому в рамках данного эксперимента не требовалось использование контейнеризации и развертывания сред. Тестирование кода осуществлялось в специальной функции-обертке,

использующей *performance API*. Результат каждой итерации сохранялся в массив. Массив с данными подвергался статистической обработке для выявления значимых различий во времени исполнения кода. Объем памяти, занимаемой средством визуализации, и объем исходных данных был получен вручную. Тип данных, принимаемых на вход функцией визуализации, был определен на этапе анализа средств визуализации.

Рассматривая все критерии в совокупности, самым оптимальным средством визуализации является *D3.js*: обладая наименьшим объемом занимаемой памяти, эта библиотека достаточно производителен и по скорости визуализации находится на втором месте после *Plotly.js*. Также она содержит в себе именно те функции, которые необходимы для осуществления поставленных целей. Бесспорно, многие из представленных инструментов обладают значительно более расширенным функционалом, но этот функционал не является необходимостью в нашем случае, а совокупность основных критериев выбора существенно понижает приоритет этих инструментов.

При разработке оптимального алгоритма визуализации данных был использован коэффициент Стьюдента для выявления закономерности из двух выборок. Таким образом, было сформулировано предположение, что выбор наиболее оптимального средства визуализации и создание алгоритма с использованием коэффициента Стьюдента позволит улучшить и ускорить процесс обработки и анализа данных, полученных при сравнении двух функций языка JavaScript.

На основании проведенных исследований был разработан следующий алгоритм визуализации данных:

1. Получение на вход двух функций языка JavaScript;
2. Определение вариантов аргументов, подаваемых на вход обеих функций;
3. Вычисление производительности в функции-обертке, использующей Performance API;
4. Получение результатов итераций в виде массивов;
5. Статистическая обработка результатов для выявления значимых различий во время исполнения кода с учётом критерия Стьюдента;
6. Визуализация полученных данных при помощи выбранного инструмента визуализации;
7. Получение на вход данных в формате CSV;
8. Предобработка исходных данных, подготовка к визуализации;
9. Вызов метода, осуществляющего отрисовку графиков и диаграмм.

Выводы. По результатам исследования были получены следующие результаты. Объем занимаемой памяти библиотек составляет: *D3.js* = 15.5 МБ, *AnyChart* = 24.69 МБ, *Chart.js* = 1.935 МБ, *Plotly.js* = 83.2 МБ, *ZingChart*: = 65.2 МБ. При измерении времени визуализации были получены результаты: *D3.js* = 2478 мс, *amCharts* = 3241 мс, *AnyChart* = 3008 мс, *Chart.js* = 4831 мс, *Google Charts* = 2508 мс, *Plotly.js* = 2283 мс, *ZingChart* = 4159 мс.

Разработан алгоритм визуализации данных, позволяющий не только визуально отобразить результаты и время выполнения двух функций JavaScript, но и выявить закономерности из двух выборок путём использования коэффициента Стьюдента.

Милантьева Н.А. (автор)

Подпись

Государев И.Б. (научный руководитель)

Подпись